MS224E

C-413 PIMag® Controller

User Manual

Version: 2.0.0 Date: 2/7/2025

This document describes the following products:

- **C-413.20**
  PIMag® motion controller; 2 channels; OEM board
- **C-413.20A**
  PIMag® motion controller; 2 channels; OEM board; analog inputs
- **C-413.2G**
  PIMag® motion controller; 2 channels; benchtop device
- **C-413.2GA**
  PIMag® motion controller; 2 channels; benchtop device; analog inputs

Physik Instrumente (PI) SE & Co. KG, Auf der Römerstraße 1, 76228 Karlsruhe, Germany
Phone +49 721 4846-0, fax +49 721 4846-1019, e-mail info@pi.ws, www.pi.ws

The following trademarks are the intellectual property of Physik Instrumente (PI) SE & Co. KG ("PI") and have been entered in the trademark register of the German Patent and Trade Mark Office and, in some cases, also in other trademark registers under the company name of Physik Instrumente (PI) GmbH & Co. KG: PI®, PIC®, PICMA®, PILine®, PIFOC®, PiezoWalk®, NEXACT®, NEXLINE®, PInano®, NanoCube®, Picoactuator®, PicoCube®, PIMikroMove®, PIMag®, PIHera®

Notes on brand names and third-party trademarks:
Microsoft® and Windows® are registered trademarks or trademarks of Microsoft Corporation in the USA and/or other countries.
EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
TwinCAT® is a registered trademark of and licensed by Beckhoff Automation GmbH.
LabVIEW, National Instruments and NI are trademarks of National Instruments. Neither the driver software nor the software programs offered by PI or other goods and services are connected to or sponsored by National Instruments.
Python® is a registered trademark of Python Software Foundation.
BiSS is a registered trademark of iC-Haus GmbH.

The following designations are protected company names, trademarks or registered trademarks of other owners:
Linux, MATLAB, MathWorks, FTDI
These designations are used for identification purposes only.

The software products provided by PI are subject to the General Software License Terms of Physik Instrumente (PI) SE & Co. KG and may contain and/or use third-party software components. Further information can be found in the General Software License Terms (https://www.physikinstrumente.com/fileadmin/user_upload/physik_instrumente/files/legal/General-Software-License-Agreement-Physik-Instrumente.pdf) and in the Third-Party Software Notes (https://www.physikinstrumente.com/fileadmin/user_upload/physik_instrumente/files/legal/Third-Party-Software-Note-Physik-Instrumente.pdf) on our website.

Original instructions
First printing: 2/7/2025
Document number: MS224E, BRo, ASt, Version 2.0.0

Subject to change. This manual is superseded by any new release. The latest respective release is available for download on our website (https://www.physikinstrumente.com/en/).

Contents

1 About this Document

1.1 Objective and Target Group of this User Manual

This user manual contains the information necessary for using the C-413 as intended.

We assume that the user has basic knowledge of closed-loop systems, motion control concepts, and applicable safety measures.

1.2 Symbols and Typographic Conventions

The following symbols and typographic conventions are used in this user manual:

NOTICE

Dangerous situation

Failure to comply could result in damage to the equipment.

➢ Precautions to avoid the risk

INFORMATION

Information for easier handling, tricks, tips, etc.

Symbol/Label	Meaning
RS-232	Label on the product indicating an operating element (example: RS-232 interface socket)
	Warning sign on the product referring to detailed information in this manual.
Start > Settings	Menu path in the PC software (example: to open the menu, the ***Start*** and ***Settings*** menu items must be selected successively)
`POS?`	Command line or a command from PI's General Command Set (GCS) (example: command to get the axis position)
Device S/N	Parameter name (example: parameter where the serial number is stored)
5	Value that must be entered or selected via the PC software

1.3 Definition of Terms

Term	Explanation
Axis	Also referred to as "logical axis". The logical axis represents the motion of the mechanics in the firmware of the C-413. For mechanics that allow motion in several directions (e.g., in X, Y, and Z), each direction of motion corresponds to a logical axis.
GCS	PI General Command Set: command set for PI controllers
Firmware	Software that is installed on the controller.
Volatile memory	RAM module where the parameters are saved when the controller is switched on (working memory). The parameter values in the volatile memory determine the current behavior of the system. The parameter values in the volatile memory are also referred to as "Active Values" in the PC software from PI.
Incremental position sensor	Sensor (encoder) for detecting changes of position or changes of angle. Signals from the incremental position sensor are used for axis position feedback. After the controller is switched on, referencing must be done before absolute target positions can be commanded and reached.
PC software	Software installed on the PC.
Nonvolatile memory	Memory module (read-only memory, e.g., EEPROM or flash memory) from which the default values of the parameters are loaded into the volatile memory when the controller is started. In the PC software from PI, the parameter values in the nonvolatile memory are also referred to as "startup values".
Mechanics	Mechanics connected to the C-413 with one or more motion axes.
Voice coil drive	A voice coil drive generates the feed via the Lorentz force on an energized coil (PIMag® principle) that is coupled to a moving rod. The drive therefore combines a relatively long travel range with a high velocity and a high resolution.

1.4 Figures

For better understandability, the colors, proportions, and degree of detail in illustrations can deviate from the actual circumstances. Photographic illustrations may also differ and must not be seen as guaranteed properties.

1.5 Other Applicable Documents

The devices and software tools from PI mentioned in this documentation are described in separate manuals.

The latest versions of the user manuals are available for download on our website (p. 3).

Description	Document
Digital Motor Controllers and Drivers	MS242EK Short Instructions
SPI Interface of the C-413	C413T0014 Technical Note
PI GCS2 driver library for use with NI LabVIEW software	SM158E Software Manual
PI GCS 2.0 DLL	SM151E Software Manual
PI MATLAB driver GCS 2.0	SM155E Software Manual
GCS array data format description	SM146E Software Manual
PIMikroMove	SM148E Software Manual
PIFirmwareManager for updating the firmware	SM164E Software Manual
PIUpdateFinder: Updating PI Software	A000T0028 Benutzerhandbuch
PI Software on ARM-Based Platforms	A000T0089 Technical Note
Downloading manuals from PI: PDF file with links to the manuals for digital electronics and software from PI. Supplied with the PI software.	A000T0081 Technical Note

1.6 Downloading manuals

INFORMATION

If a manual is missing or problems occur with downloading:
- Contact PI's customer service (p. 287).

Downloading manuals

1. Open the website **www.pi.ws**.
2. Search the website for the product number (e.g., C-413).
3. In the search results, select the product to open the product details page.
4. Select ***Downloads***.

   Manuals are shown under ***Documentation***. Software manuals are shown under ***General Software Documentation***.
5. For the desired manual, select ***ADD TO LIST*** and then ***REQUEST***.

6. Fill out the request form and select ***SEND REQUEST***.

   The download link will then be sent to the email address entered.

2 Safety

2.1 Intended Use

The C-413 is a laboratory device as defined by DIN EN 61010-1. It is intended for indoor use and use in an environment that is free of dirt, oil, and lubricants.

The C-413.20A and .20 OEM boards must be installed in a suitable housing before startup.

In accordance with its design, the C-413 is intended to be used for operating mechanics with voice coil drives (p. 2).

The C-413 is intended for closed-loop operation. For closed-loop operation, sensor signals must be provided via an SPI interface or optionally as analog input signals. Furthermore, the C-413 can read out the reference switch signals of the mechanics and process them further.

The C-413 may only be used in compliance with the technical specifications and instructions in this user manual. The user is responsible for process validation.

If an electrical operating device is designed to be integrated into another electrical operating device: The operator is responsible for standards compliant integration of the electrical device into the overall system.

2.2 General Safety Instructions

The C-413 is built according to state-of-the-art technology and recognized safety standards. Improper use of the C-413 may result in personal injury and/or damage to the C-413.

- Use the C-413 for its intended purpose only, and only when it is in perfect condition.
- Read the user manual.
- Immediately eliminate any faults and malfunctions that are likely to affect safety.

The operator is responsible for installing and operating the C-413 correctly.

- Install the C-413 near the power source so that the power plug can be quickly and easily disconnected from the mains.
- Use the supplied components (power supply, adapter, power cord) to connect the C-413 to the power source.
- If one of the supplied components for connecting to the power source has to be replaced, use a sufficiently dimensioned component.

2.3 Organizational Measures

User manual

- Keep this user manual with the C-413 always.
  The latest versions of the user manuals are available for download on our website (p. 3).
- Add all manufacturer information such as supplements or technical notes to the user manual.
- If you give the C-413 to other users, also include this user manual as well as other relevant information provided by the manufacturer.
- Always work according to the complete user manual. The equipment can be damaged if your user manual is incomplete and is therefore missing important information.
- Install and operate the C-413 only after you have read and understood this user manual.

Personnel qualification

The C-413 may only be installed, started up, operated, maintained, and cleaned by authorized and appropriately qualified personnel.

2.4 European Declarations of Conformity

For the C-413, declarations of conformity were issued according to the following European statutory requirements:

EMC Directive

RoHS Directive

The standards applied for certifying conformity are listed below.

EMC: EN 61326-1

Safety: EN 61010-1

RoHS: EN IEC 63000

3 Product Description

3.1 Model Overview

The C-413 is available in the following versions:

Model	Name
C-413.2G	PIMag® motion controller; 2 channels; benchtop device; USB and SPI interface; force control option
C-413.2GA	PIMag® motion controller; 2 channels; benchtop device; USB and SPI interface; analog inputs; force control option
C-413.20	PIMag® motion controller; 2 channels; OEM board; USB and SPI interface; force control option
C-413.20A	PIMag® motion controller; 2 channels; OEM board; USB and SPI interface; analog inputs; force control option

3.2 Product View

3.2.1 Front View

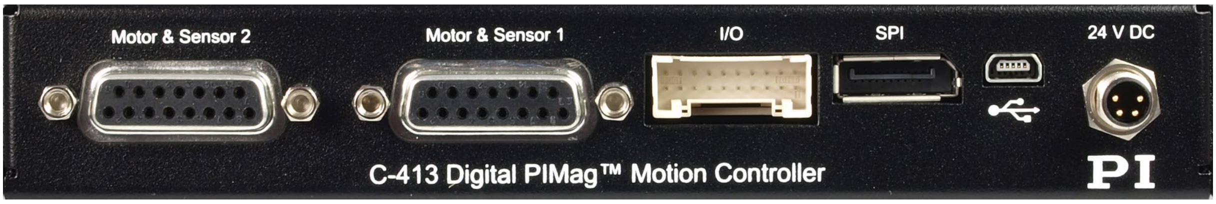


Figure 1: C-413.2GA and .2G; front view

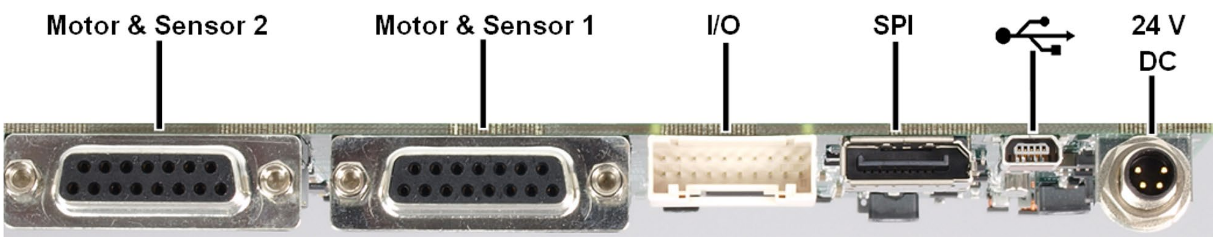


Figure 2: C-413.20A and .20; front view

Labeling	Type	Function
Motor & Sensor 1	D-sub 15 (f) (p. 294)	Connector for PI mechanics with voice coil drive and sensor: ▪ Output current for output signal channel 1 ▪ Sensor and ID chip data for input signal channels 1 and 2 (via SPI) ▪ Reference switch signals for input signal channels 1 and 2
Motor & Sensor 2	D-sub 15 (f) (p. 294)	Connector for PI mechanics with voice coil drive and sensor: ▪ Output current for output signal channel 2 ▪ Sensor and ID chip data for input signal channels 3 and 4 (via SPI) ▪ Reference switch signals for input signal channels 3 and 4
I/O	PUD panel plug (m), 20-pin (JST) (p. 295)	Digital lines: ▪ Outputs: Triggering of external devices, output of the servo cycles ▪ Inputs: Triggering of data recorder or wave generator Only C-413.2GA and .20A - analog lines: ▪ Inputs: Used for external sensors or as analog control inputs ▪ Outputs: Used to monitor the position, force or velocity of an axis or for controlling external motor drivers
SPI	Display port	Connector for SPI master (**S**erial **P**eripheral **I**nterface). For transmission of current values and target/control values between the C-413 and the SPI master with minimum latency and high update rate. ASCII data can also be transmitted so that the SPI master has full access to the PI General Command Set (GCS). Refer to the C413T0014 technical note for details.
	Mini USB type B 	Universal serial bus for connecting to the PC
24 V DC	M8 panel plug, 4-pin (p. 297)	Connector for the supply voltage

3.2.2 Type Plate

Figure 3: C-413.2GA: Type plate on the top side

Labeling	Function
	Data matrix code (example; contains the serial number)
C-413	Product name (example), the characters following the period refer to the model
	Manufacturer's logo
113064443	Serial number (example), individual for each C-413 Meaning of each position (from the left): 1 = internal information, 2 and 3 = year of manufacture, 4 to 9 = consecutive number
24 V / 2 A DC	Operating voltage / current consumption
Country of origin: Germany	Country of origin
	Warning sign "Pay attention to the manual!"
	Old equipment disposal (p. 299)
WWW.PI.WS	Manufacturer's address (website)
	CE conformity mark

3.2.3 Protective Earth Connection

C-413.2GA and .2G models

Figure 4: Protective earth connection of the C-413.2GA and .2G models

Labeling	Type	Function
⏚	M4 threaded pin	Protective earth connection

C-413.20A and .20 models

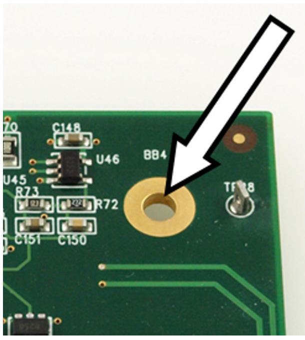


Figure 5: One of four mounting holes of the C-413.20A and .20 models

Labeling	Type	Function
-	Hole with Ø 2.8 mm, 4 x present	Mounting holes with GND potential Protective earth connection via the electrical connection of the mounting holes with a housing connected to the protective earth conductor (p. 61).

3.3 Scope of Delivery

Item	Component
C-413	Model according to your order
K050B0003	Adapter for the power supply connection; barrel connector to M8 4-pin connector (f)
C-990.CD1	Data storage device with PC software from PI
MS242EK	Short instructions for digital motor controllers and drivers
Only with the C-413.2G and .2GA benchtop devices:	
C-501.24050H	Wide-range-input power supply 24 V DC / 50 W
3763	Power cord
000036360	USB cable (type A to mini-B) for connection to the PC, 3 m

3.4 Optional Accessories

Order number	Description
C-413.1IO	I/O cable for C-413 PIMag® Motion controller, 1 m, open end Refer to the "C-413.1IO Cable for the I/O Connection" (p. 296) for details.

To order, contact our customer service (p. 287).

3.5 Functional Principles

3.5.1 Block Diagram

The C-413 is intended for the operation of two logical axes that are each equipped with a voice coil drive, incremental position sensor and - optionally - with a force sensor.

The following block diagram shows how the C-413 generates the output current for an axis:

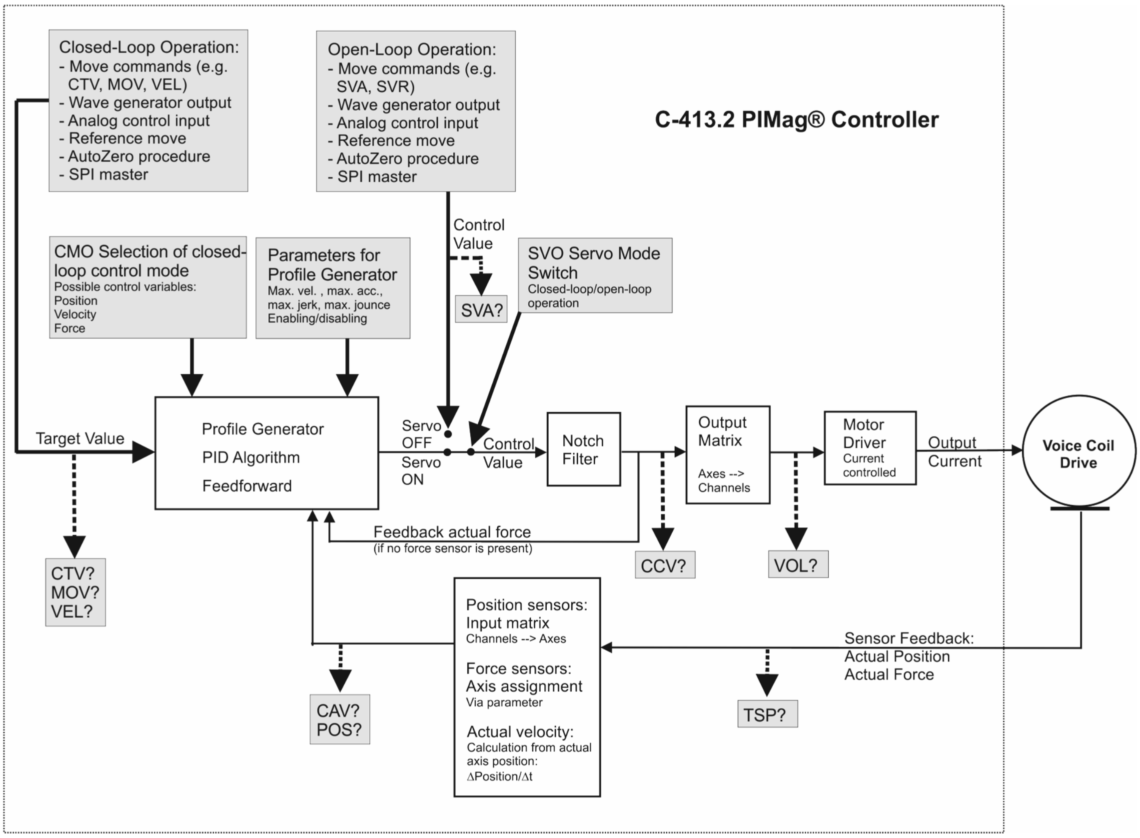


Figure 6: C-413: Generation of the output current for one axis

The following diagram shows the relationships between the input and output signals and the logical axes of the C-413.

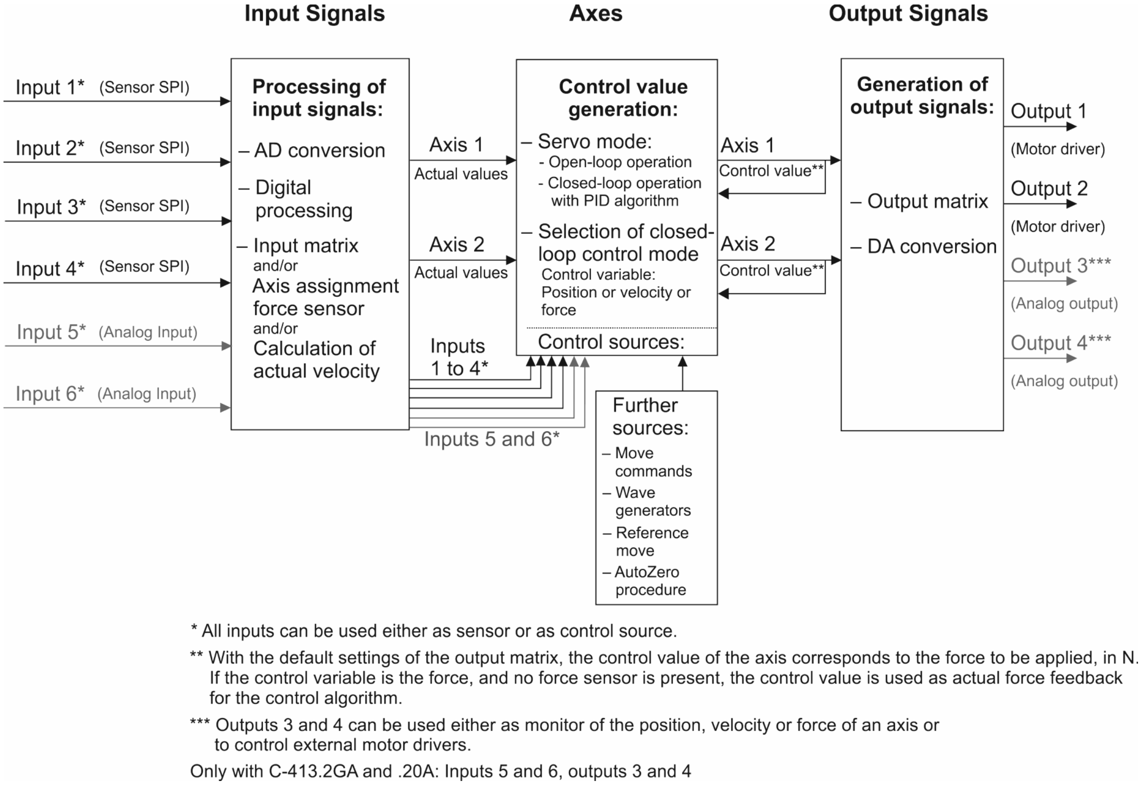


Figure 7: C-413: Relationships between channels and axes of the C-413

The components of the diagrams are described in more detail in the following sections.

3.5.2 Commandable Elements

The following table contains the elements that can be commanded with GCS commands (p. 148).

Element	Number	Identifier	Description
Logical axis	2	1, 2	The logical axis represents the motion of the mechanics in the firmware of the C-413. It corresponds to an axis of a linear coordinate system. All commands for the motion of the mechanics refer to logical axes. The value of the ***Number Of System Axes*** parameter (ID 0x0E000B02) specifies the number of axes. The input and output signal channels of the C-413 can be flexibly allocated to the axes (p. 16). Axes can be "deactivated" (p. 52).

Element	Number	Identifier	Description
Input signal channels	4 or 6	1 to 4 or 1 to 6	1, 2, 3, 4: channels for sensors; inputs via the **Motor & Sensor** (p. 294) sockets**.** The number can be queried via the ***Number Of Sensor Channels*** parameter (ID 0x0E000B03). 5 and 6 only with C-413.2GA and .20A: Analog input channels; input via the **I/O** panel plug (p. 295). Each analog input can be used for an external sensor or as a control source. The total number of input signal channels can be queried via the ***Number Of Input Channels*** parameter (ID 0x0E000B00).
Output signal channels	2 or 4	1, 2 or 1 to 4	1 and 2: Motor driver outputs channels of the C-413; output via the **Motor & Sensor** (p. 294) sockets**.** The number can be queried via the ***Number Of Driver Channels*** parameter (ID 0x0E000B04). 3 and 4 only with C-413.2GA and .20A: Analog output channels; output via the **I/O** panel plug (p. 295). Each analog output can be used to monitor the position, force or velocity of an axis or to control an external motor driver. The total number of output signal channels can be queried via the ***Number Of Output Channels*** parameter (ID 0x0E000B01).
Digital outputs	5	1 to 5	1 to 5 identify digital output lines 1 to 5 of the **I/O** panel plug (p. 295). Digital output 6 (pin 20 of the **I/O** panel plug) is not accessible for commands. It outputs the servo cycles. Refer to "Digital Output Signals" (p. 85) for further information.
Digital inputs	4	1 to 4	1 to 4 identify digital input lines 1 to 4 of the **I/O** panel plug (p. 295). Refer to "Digital Input Signals" (p. 90) for further information.
Wave generator	2	1, 2	The number of wave generators corresponds to the number of logical axes. Each wave generator is permanently allocated to a logical axis (p. 120).
Wave table	8	1 to 8	The wave tables contain the saved data (a total of 4096 points) for the waveforms that are output by the wave generators. The value of the ***Number Of Waves*** parameter (ID 0x1300010A) indicates the number of wave tables (p. 120).
Data recorder table	≤8	1, 2, ...	The data recorder tables contain the recorded data (a total of 4096 points, refer to "Data Recorder" (p. 83)). The number of data recorder tables can be set with the ***Data Recorder Chan Number*** parameter (ID 0x16000300). The ***Max Number Of Data Recorder Channels*** parameter (ID 0x16000100) indicates the maximum number of data recorder tables.

Element	Number	Identifier	Description
Overall system	1	1	C-413 as an overall system.

3.5.3 Important Components of the Firmware

The firmware of the C-413 provides the following functional units:

Firmware Component	Description
ASCII commands	Communication with the C-413 can be managed using the commands of the PI General Command Set (GCS; version 2.0). GCS is independent of the hardware (controller, mechanics). Examples of the use of GCS: ▪ Configuring the C-413 ▪ Switching the servo mode on and off ▪ Starting axis motion ▪ Getting values You can find a list of the available commands in the "Command Overview" section (p. 145).
Parameters and command levels	Parameters reflect the properties of the C-413 and the mechanics connected, and define the behavior of the system (e.g., settings for matrices (p. 16), selecting control mode, and settings for the servo algorithm). The parameters can be divided into the following categories: ▪ Protected parameters whose default settings cannot be changed ▪ Parameters that can be set by the user to adapt to the application Write permission for the parameters is determined by command levels. The current command level can be changed with the `CCL` command. This may require entering a password. Refer to "Adapting Settings" (p. 257) for further information. The values of some parameters are stored on the ID chip (p. 53) of the mechanics. They are written to the volatile and the nonvolatile memory when switching on or rebooting the C-413.
Control modes and control variables	The control mode for closed-loop operation can be selected. The selected control mode determines the control variable. Target value specifications refer to the control variable, which can be one of the following: ▪ Position ▪ Velocity ▪ Force Depending on the control mode, the "inner" control variables of the position and/or velocity are controlled in addition to the actual control

Firmware Component	Description
	variable. Refer to "Control Modes and Control Variables" (p. 28) for further information.
Profile generator, servo algorithm, feedforward	In closed-loop operation, a profile generator calculates the dynamics profile for the control variable from the target value (default setting; profile generator can be deactivated). The error resulting from the difference between the target value or calculated dynamics profile and the current value of the control variable runs through a PID servo algorithm. Feedforward can be used optionally to improve the tracking performance and minimize the following error. You can find further information in the sections "Generating a Dynamics Profile" (p. 30) and "Servo Algorithm and Other Control Value Corrections" (p. 33).
Wave generator	Each axis can be controlled by a wave generator that outputs waveforms. The wave generator is especially suited for dynamic applications in which periodic motion of the axis is executed (p. 120).
Data recorder	The C-413 contains a real-time data recorder (p. 83). This can record different input and output signals (e. g. current position, sensor input, output current) from different data sources (e.g., logical axes, input and output signal channels).

The firmware can be updated with a tool (p. 279).

3.5.4 Allocating Axes to Channels

In the firmware of the C-413, input signal channels and output signal channels can be flexibly allocated to the logical axes.

- Allocation via matrices:
  - Input signal channel-to-axis matrix, "input matrix" (p. 16) for short; intended for the allocation of position sensors and - with C-413.2GA and .20A - the additional analog inputs to axes
  - Axis-to-output signal channel matrix, "output matrix" (p. 19) for short; intended for allocating motor driver outputs and - with C-413.2GA and .20A - the additional analog outputs to axes

  The matrices also define how much the channels contribute to measuring the axis position or controlling the drives of the axis.
- Direct allocation of force sensors to axes (p. 19)

Input matrix

Up to 4 (C-413.2G, .20) or 6 (C-413.2GA, .20A) sensors can be used to monitor the axis position:

- Input signal channels 1 to 4 on the **Motor & Sensor** sockets (p. 294) (sensor data via SPI)
- C-413.2GA and .20A only: Input signal channels 5 and 6 on the **I/O** panel plug (p. 295) (analog inputs for sensors, can also be used as analog control inputs (p. 94))

The axis positions are calculated via the input matrix from the position values of the input signal channels.

Input matrix for C-413.2G and .20

$$\begin{pmatrix} \text{Axis position 1} \\ \text{Axis position 2} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{pmatrix} \begin{pmatrix} \text{Input 1} \\ \text{Input 2} \\ \text{Input 3} \\ \text{Input 4} \end{pmatrix}$$

Written as equations:

Axis position 1 = a_{11}•*input 1* + a_{12}•*input 2* + a_{13}•*input 3* + a_{14}•*input 4*

Axis position 2 = a_{21}•*input 1* + a_{22}•*input 2* + a_{23}•*input 3* + a_{24}•*input 4*

Input matrix for C-413.2GA and .20A

$$\begin{pmatrix} \text{Axis position 1} \\ \text{Axis position 2} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \end{pmatrix} \begin{pmatrix} \text{Input 1} \\ \text{Input 2} \\ \text{Input 3} \\ \text{Input 4} \\ \text{Input 5} \\ \text{Input 6} \end{pmatrix}$$

Written as equations:

Axis position 1 = a_{11}•*input 1* + a_{12}•*input 2* + a_{13}•*input 3* + a_{14}•*input 4* + a_{15}•*input 5* + a_{16}•*input 6*

Axis position 2 = a_{21}•*input 1* + a_{22}•*input 2* + a_{23}•*input 3* + a_{24}•*input 4* + a_{25}•*input 5* + a_{26}•*input 6*

The coefficients of the input matrix are set by PI before delivery and are defined by the following parameters, whereby i stands for the identifier of the axis and can take on the values 1 and 2.

Coefficient	Parameters	Description
a_{i1}	***Position From Sensor 1*** ID 0x07000500	Coefficient for sensor 1 for calculating the position of axis i
a_{i2}	***Position From Sensor 2*** ID 0x07000501	Coefficient for sensor 2 for calculating the position of axis i
a_{i3}	***Position From Sensor 3*** ID 0x07000502	Coefficient for sensor 3 for calculating the position of axis i
a_{i4}	***Position From Sensor 4*** ID 0x07000503	Coefficient for sensor 4 for calculating the position of axis i
a_{i5}	***Position From Sensor 5*** ID 0x07000504	Only C-413.2GA and .20A; coefficient for sensor 5 for calculating the position of axis i

Coefficient	Parameters	Description
a_{i6}	***Position From Sensor 6*** ID 0x07000505	Only C-413.2GA and .20A; coefficient for sensor 6 for calculating the position of axis i

A matrix coefficient must be zero in the following cases:

- No sensor is connected at the input signal channel.
- A force sensor is connected at the input signal channel (p. 106).
- The input signal channel is used as the control source (p. 107).

INFORMATION

You can display an overview of the matrix coefficients in PIMikroMove:

1. Open the ***Device Parameter Configuration*** window via the ***C-413 > Parameter Configuration...*** menu item.
2. Open the ***Axis Matrices*** window by selecting the ***View > Axis Matrices*** menu item in the ***Device Parameter Configuration*** window.

INFORMATION

The current position values can be queried as follows:

- Input signal channels: `TSP?` command
- Axes: `POS?` command. When the control variable is the position (selection of the control mode, see `CMO` (p. 159)), the `CAV?` command can also be used to get the position.

INFORMATION

When the control variable is the velocity (selection of the control mode, see `CMO` (p. 159)):

- The current velocity of the axis is calculated from the current position value of the axis: current velocity = Δ current position / Δ time
- The current velocity can be queried with the `CAV?` command.

INFORMATION

The input matrix is only intended for position sensors. When a force sensor is used, the corresponding input signal channel must be allocated to the axis via the ***Input Channel for Force Feedback*** parameter (ID 0x07000400) (p. 19).

INFORMATION

The settings of the input matrix are also used to generate the response to the axis-related query of the mechanics name from the channel-related parameters 0x0F000100 (see `CST?` (p. 162)).

Direct allocation of force sensors to axes

To monitor the force generated by the axes, one force sensor can be used for each axis. Connection options on the C-413:

- Input signal channels 1 to 4 on the **Motor & Sensor** sockets (p. 294) (sensor data via SPI)
- Only C-413.2GA and .20A: Input signal channels 5 and 6 on the **I/O** panel plug (p. 295) (analog inputs for sensors, can also be used as analog control inputs (p. 94))

Force sensors are directly allocated to the logical axes of the C-413 via the ***Input Channel For Force Feedback*** parameter (ID 0x07000400).

When a force sensor is connected to an input signal channel, the coefficients of the input matrix (p. 16) must be zero for this channel.

INFORMATION

The current force value measured by the force sensor can be queried as followed:

- Input signal channels: `TSP?` command
- Axes: When the control value is the force, (to select the control mode, see `CMO` (p. 159)), the `CAV?` command can be used to get the current force.

When no force sensor is present: With the default settings of the output matrix (p. 19), the control value of the axis is the force to be generated in N. The control value of the axis can be queried with the `CCV?` command (p. 158).

Output matrix

Up to 2 (C-413.2G, .20) or 4 (C-413.2GA, .20A) drives can be controlled to move the axes. The following output signal channels of the C-413 are available for controlling drives:

- Output signal channels 1 and 2 on the **Motor & Sensor** sockets (p. 294) (motor driver outputs)
- Only C-413.2GA and .20A: Output signal channels 3 and 4 on the **I/O** panel plug (p. 295) (analog outputs for external motor drivers, can also be used to monitor the axis position (p. 110))

The control values of the axes are converted to the output values of the output signal channels via the output matrix.

Output matrix for C-413.2G and .20

$$\begin{pmatrix} \mathit{Output\ 1} \\ \mathit{Output\ 2} \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} \begin{pmatrix} \mathit{Axis\ 1} \\ \mathit{Axis\ 2} \end{pmatrix}$$

Written as equations:

Output 1 = p_{11} • *axis 1* + p_{12} • *axis 2*

Output 2 = p_{21} • *axis 1* + p_{22} • *axis 2*

Output matrix for C-413.2GA and .20A

$$\begin{pmatrix} \mathit{Output\ 1} \\ \mathit{Output\ 2} \\ \mathit{Output\ 3} \\ \mathit{Output\ 4} \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ p_{31} & p_{32} \\ p_{41} & p_{42} \end{pmatrix} \begin{pmatrix} \mathit{Axis\ 1} \\ \mathit{Axis\ 2} \end{pmatrix}$$

Written as equations:

Output 1 = p_{11} • *axis 1* + p_{12} • *axis 2*

Output 2 = p_{21} • *axis 1* + p_{22} • *axis 2*

Output 3 = p_{31} • *axis 1* + p_{32} • *axis 2*

Output 4 = p_{41} • *axis 1* + p_{42} • *axis 2*

The coefficients of the output matrix are set by PI before delivery and are defined by the following parameters, whereby i stands for the identifier of the axis and can take on the values 1 and 2:

Coefficient	Parameter	Description
p_{1i}	***Driving Factor 1*** ID 0x09000000	Coefficient of axis i for the output value at output 1
p_{2i}	***Driving Factor 2*** ID 0x09000001	Coefficient of axis i for the output value at output 2
p_{3i}	***Driving Factor 3*** ID 0x09000002	Only C-413.2GA and .20A; coefficient of axis i for the output value at output 3
p_{4i}	***Driving Factor 4*** ID 0x09000003	Only C-413.2GA and .20A; coefficient of axis i for the output value at output 4

INFORMATION

You can display an overview of the matrix coefficients in PIMikroMove:

1. Open the ***Device Parameter Configuration*** window via the ***C-413 > Parameter Configuration...*** menu item.
2. Open the ***Axis Matrices*** window by selecting the ***View > Axis Matrices*** menu item in the ***Device Parameter Configuration*** window.

INFORMATION

The current output values can be queried as follows:

- Output signal channels: `VOL?` command
- Axes (control value) `CCV?` command

INFORMATION

The matrix coefficients for output signal channels 1 and 2 are numerically set by PI to the current value that the respectively connected drive needs to generate a force of 1 N (unit: A/N). The control value of an axis thus corresponds to the force in N to be generated.

3.5.5 Processing of Input Signal Channels

After the A/D conversion (only channels 5 and 6 with C-413.2GA and .20A), the digital signals of the input signal channels are processed further. The processing consists of the following steps:

- Electronics linearization
- Mechanics linearization
- Digital filtering

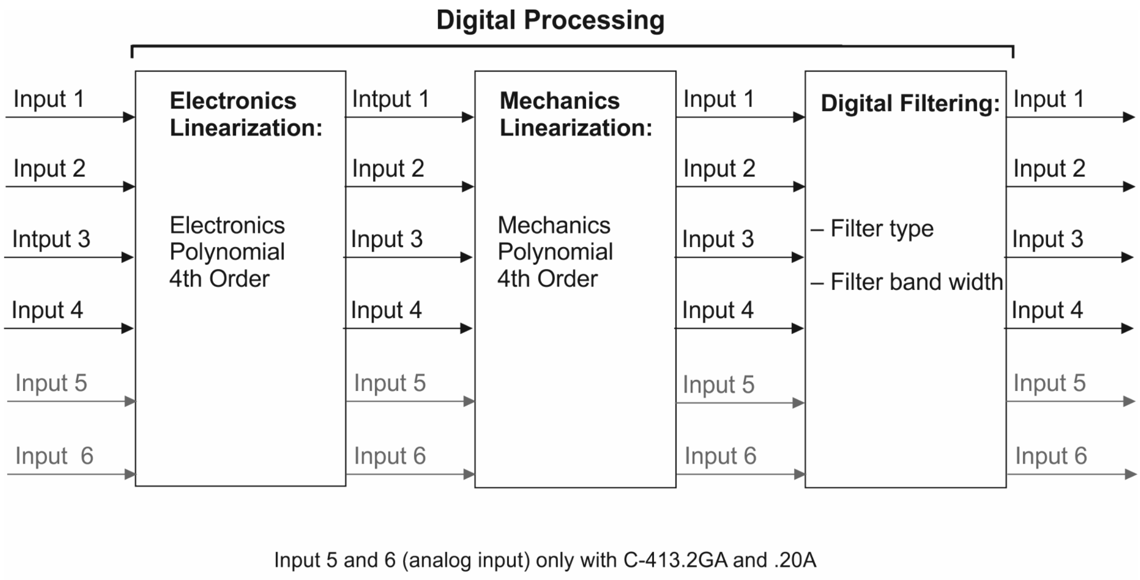


Figure 8: Digital processing of input signals

Linearization

Polynomial linearization is used to improve the system performance.

Basic form of the used polynomials:

$y = a_0 + a_1 \bullet x + a_2 \bullet x^2 + a_3 \bullet x^3 + a_4 \bullet x^4$

x: Digital, filtered value of the input signal channel
y: Linearized value of the input signal channel

Different polynomials are used for linearizing the mechanics and the electronics in order to simplify the replacement of system components. The coefficients of the polynomials are defined by PI.

Terms	Parameters of the Coefficients	Description
Electronics:		
▪ Offset	ID 0x03000100	When an incremental sensor is used for the input signal channel, the gain must be set to the inverted value of the interpolation rate of the sensor. The offset must be zero. The result after the linearization (normalized value without a unit) can be queried with the `TNS?` command.
▪ Gain	ID 0x03000200	
▪ 2nd order correction	ID 0x03000300	
▪ 3rd order correction	ID 0x03000400	
▪ 4th order correction	ID 0x03000500	
Mechanics:		
▪ Offset	ID 0x02000200	The settings are loaded into the C-413 from the ID chip (p. 53) of the mechanics. For the input signal channels 5 and 6 (only with C-413.2GA and .20A), the offset and gain must be set to scale the analog inputs to suitable values (p. 98). When a force sensor is used, the autozero procedure adjusts the offset coefficient of the corresponding input signal channel (p. 49). The result after the linearization (scaled value in physical units) can be queried with the `TSP?` command.
▪ Gain	ID 0x02000300	
▪ 2nd order correction	ID 0x02000400	
▪ 3rd order correction	ID 0x02000500	
▪ 4th order correction	ID 0x02000600	

Digital filtering

Parameters	Function
Digital Filter Type ID 0x05000000	Filter type ▪ 0: No filtering ▪ 1: IIR second-order low-pass filter
Digital Filter Bandwidth ID 0x05000001	Frequency of the IIR low-pass filter (only for ***Digital Filter Type*** = 1)

3.5.6 Servo Modes

The servo mode determines whether the motion is performed in closed-loop operation or in open-loop operation.

Operating mode	Description
Closed-loop operation (servo mode On)	The control variable is selected by selecting the control mode, refer to "Control Modes and Control Variables" (p. 28). In control modes 7, 9, 10, and 11, the "inner" (actual) control variables of the position and/or velocity are controlled in addition to the "outer" control variable. The control sources define the target value for the actual control variable, refer to "Generating Control Values" (p. 24). A profile generator calculates the dynamics profile for the actual control variable from the target value (default setting; profile generator can be deactivated), refer to "Generating a Dynamics Profile" (p. 30). The error that results from the difference between the target value or calculated dynamics profile and the current value of the control variable runs through a PID servo algorithm. Feedforward can be used optionally to improve the tracking performance and minimize the following error. The result is the control value of the axis, refer to "Servo Algorithm and Other Control Value Corrections" (p. 33).
Open-loop operation (servo mode Off)	The control sources directly specify the control value of the axis, refer to "Generating Control Values" (p. 24). In open-loop operation, the C-413 does **not** evaluate the current values for position, velocity, and force.

INFORMATION

After the C-413 is switched on or rebooted, open-loop operation is activated by default (servo mode Off).

- Get the current operating mode with the `SVO?` or `SRG?` commands.
- Enable closed-loop operation with the `SVO` command.
- If necessary configure the C-413 with the ***Power Up Servo Enable*** parameter (ID 0x07000800) so that servo mode is switched on automatically when switching on or rebooting.

INFORMATION

In the following cases, it is not permitted to switch servo mode on or off:

- The wave generator is running for the axis.
- An input signal channel is used as the control source for the axis.

3.5.7 Generating Control Values

The axis motion can be triggered by various control sources. The control sources specify the following, depending on the servo mode (p. 22):

- Closed-loop operation: Target value of the control variable (p. 28)

  The target value is included in the servo algorithm (p. 33) either directly or via the profile generator (p. 30). The result of the servo algorithm (PID with optional feedforward) is the control value of the axis.

- Open-loop operation: Control value of the axis

The control value of the axis is corrected by notch filters (p. 33) and is converted to the output values of the output signal channels by the output matrix (p. 19).

INFORMATION

The matrix coefficients for output signal channels 1 and 2 are numerically set by PI to the current value that the respectively connected drive needs to generate a force of 1 N (unit: A/N). The control value of an axis thus corresponds to the force in N to be generated.

Control sources

The control sources have different priorities, i. e. control sources can overwrite the specifications of other control sources.

The following table lists the control sources according to their priority:

- First line: Lowest priority, is overwritten by all other control sources
- Last line: Highest priority, overwrites all other control sources.

Control source	Commands	Function

Motion commands	`CTV`
	`CTR`
	`IMP`
	`STE`
		Specify the target value of the control variable in closed-loop operation. Depending on the selected control mode, `MOV`, `MVR` or `VEL` can also be used.

	`SVA`
	`SVR`
	`IMP`
	`STE`
		Specify the control value in open-loop operation.

Wave generator (p. 120)	`WGO`
	
	`WOS`
		Switches on the wave generator for periodic motions.
		Depending on the servo mode, the wave generator outputs target values or control values.
		An offset can be added to the output value of the wave generator with the `WOS` command.
		An input signal channel can be configured as a control source for an axis while the wave generator is active for this axis. In that case, the wave generator will continue running, but its output will no longer be used as a target or control value.

Control source	Commands	Function
Input signal channel, also called an "analog control input" (p. 107)	SPA, SEP	When an input signal channel is active as the control source for an axis, it specifies target or control values, depending on the servo mode. In order to use an input signal channel as a control source for the axis, it must be connected to the axis. For this purpose, the ***ADC Channel For Target*** parameter (ID 0x06000500) is correspondingly set with the SPA or SEP command. In addition, the coefficients of the input matrix (p. 16) must be set to 0 for the selected input signal channel. The AOS command can be used to add an offset to the value of the input signal channel connected. When an input signal channel is used as a control source and the axis motion is stopped with the STP or #24 command, the connection between the channel and the axis is terminated. To command the axis via the input signal channel again, the input signal channel must be reconnected to the axis.
Referencing move (p. 46)	FRF	Closed-loop operation: Sets the target value to zero. Open-loop operation: Sets the control value to the value of the ***AutoZero Result*** parameter (0x07000A03) at the end of the referencing move.
Autozero procedure (p. 49)	ATZ	Closed-loop operation: ▪ Position as control variable: Sets the target value to the current position at the end of the autozero procedure. ▪ Velocity or force as control variable: Sets the target value to zero. Open-loop operation: Sets the control value to the value at which the generated force is 0 N at the end of the autozero procedure. When an input signal channel is active as the control source, it is automatically deactivated at the start of the autozero procedure with the ATZ command and reactivated again after the procedure.

INFORMATION

The C-413 can also be controlled by an SPI master. The SPI master transmits either target/control values or GCS commands, depending on the data segment used. The C-413 handles the target/control values received by the SPI master in the same way as input via an analog control input. Refer to the C413T0014 Technical Note for details.

The target or control value of an axis is also influenced by the following actions:

Action	Commands	Effect
Setting of the servo mode	`SVO`	When the servo mode is switched on, the target value for the control variable is set as follows: ▪ Position as control variable: The target value is set to the current value of the position. ▪ Velocity or force as control variable: The target value is set to zero. When the servo mode is switched off, the control value is set to the value of the ***AutoZero Result*** parameter (ID 0x07000A03).
Selecting the control mode	`CMO`	When the control mode is changed, the target value for the control variable is set as follows: ▪ The new control variable is the position: The target value is set to the current value of the position. ▪ The new control variable is the velocity or the force: The target value is set to zero.

INFORMATION

In the following cases, it is not permitted to switch servo mode on or off or to change the control mode:

- The wave generator is running for the axis.
- An input signal channel is used as the control source for the axis.

INFORMATION

In control modes 7, 9, 10, and 11 (p. 28), the "inner" control variables of the position and/or velocity are controlled in addition to the "outer" (actual) control variables. The target value for the control variables in these control modes is defined as follows:

- Actual control variable: Target value specification by the control sources
- Internal control variables: Target value specification by the outer servo loop

Permissible range for the target value and control value

The following parameters limit the permissible range for the target value and control value:

Parameters	Description and Possible Values
Profile Generator Maximum Velocity 0x06010400	Maximum velocity for closed-loop operation Limits the velocity that can be set with `VEL`. Limits the velocity of the dynamics profile when the control variable is the position. Further use when the control variable is the position or force: The velocity is set to the value of the parameter when the C-413 is switched

Parameters	Description and Possible Values
	on or rebooted and when you switch from velocity control to position or force control. Changing the parameter value in the volatile memory overwrites the velocity currently set with `VEL`.
Position Range Limit Min 0x07000000	Minimum commandable position in closed-loop operation The value of the parameter in the volatile memory can also be queried with the `TMN?` command. When the control variable is the position, the value can also be queried with the `CMN?` command.
Position Range Limit Max 0x07000001	Maximum commandable position in closed-loop operation The value of the parameter in the volatile memory can also be queried with the `TMX?` command. When the control variable is the position, the value can also be queried with the `CMX?` command.
Force Range Limit min 0x07000005	Minimum commandable force in closed-loop and open-loop operation The parameter also specifies the smallest permissible value for the control value. If the control value falls below the limit value in closed-loop operation, the overflow status of the axis occurs. Note that the output is also limited by the ***Soft Limit min*** parameter (ID 0x0C000000), which specifies the smallest permissible value for an output channel signal (in A or V). When the control variable is the force, the value of the ***Force Range Limit min*** parameter in the volatile memory can be queried with the `CMN?` command.
Force Range Limit max 0x07000006	Maximum commandable force in closed-loop and open-loop operation The parameter also specifies the largest permissible value for the control value. If the control value exceeds the limit value in closed-loop operation, the overflow status of the axis occurs. Note that the output is also limited by the ***Soft Limit max*** parameter (ID 0x0C000001), which specifies the largest permissible value for an output channel signal (in A or V). When the control variable is the force, the value of the ***Force Range Limit max*** parameter in the volatile memory can be queried with the `CMX?` command.

INFORMATION

The behavior of the C-413 when the permissible range for the target value is exceeded depends on the control source used:

- Motion commands with impermissible specifications are ignored and a corresponding error code is set.
- Wave generator and input signal channel: The motion is performed and the corresponding limit value is used as the target value. An error code is not set.

INFORMATION

The currently valid limits for closed-loop operation can be queried with CMN? (p. 159) and CMX? (p. 161).

3.5.8 Control Modes and Control Variables

The control mode for closed-loop operation can be selected. The selection of the control mode determines the control variable.

The following table lists the control modes supported by C-413 and the corresponding control variables. Meaning of the markings:

- Bold: Control mode selected by default
- Gray background: Control modes selectable by default

ID	Short designation	Control mode	Control variable	Supported motion commands in closed-loop operation
1	PID_Pos	Direct PID position control	Position	MOV, MVR, CTV, CTR, STE, IMP
6	PID_Vel	Direct PID velocity control	Velocity	VEL, CTV, CTR, STE, IMP
7	**PID_Pos_Vel**	**PID position control with velocity control**	**Position**	**MOV, MVR, CTV, CTR, STE, IMP**
8	PID_Force	Direct PID force control	Force	CTV, CTR, IMP, STE
9	PID_Force_Pos	PID force control with position control	Force	CTV, CTR, IMP, STE
10	PID_Force_Pos_Vel	PID force control with position and velocity control	Force	CTV, CTR, IMP, STE
11	PID_Force_Vel	PID force control with velocity control	Force	CTV, CTR, IMP, STE

INFORMATION

The servo algorithm has a cascade structure in control modes 7, 9, 10, and 11. In the case of a cascade structure, the "inner" control variables of the position and/or velocity are controlled in addition to the "outer" (actual) control variable. The target value for inner control variables is specified by the corresponding outer servo loop. Refer to "Servo Algorithm and Other Control Value Corrections" (p. 33) for further information.

The following commands are available for selecting the control mode:

Command	Syntax	Function
CMO	CMO {<AxisID> <CtrlMode>}	Selects the control mode for closed-loop operation; for IDs for <CtrlMode>, refer to the table above. Sets the value of the ***Closed-Loop Control Mode*** parameter (ID 0x07030100) in the volatile memory. Required command

Command	Syntax	Function
		level: 0.
`CMO?`	CMO? [{<AxisID>}]	Queries the selected control mode for closed-loop operation (value of the ***Closed-Loop Control Mode*** parameter in the volatile memory).

The selection of the control mode can be configured with the following parameters:

<table>
<tr><th>Parameters</th><th>Description and Possible Values</th></tr>
<tr><td>Closed-Loop Control Mode
0x07030100</td><td>Selected control mode for closed-loop operation
Possible values: Refer to the IDs in the above table of control modes, limited by the Available Closed-Loop Control Modes parameter.
Setting with the CMO or SPA (p. 199) command. Setting with SPA requires switching to command level 1 with the CCL command (p. 157).</td></tr>
<tr><td>Available Closed-Loop Control Modes
0x07030101</td><td>Selectable control modes for closed-loop operation
Limits the control modes supported by C-413 to actual selectable control modes (limited to value range of the Closed-Loop Control Mode parameter). This is to prevent inadvertently selecting a control mode where the servo control parameters of the C-413 are not adapted. The value of the parameter is bit-coded and includes the IDs of the currently selectable control modes. Examples:
<table>
<tr><td>IDs of the selectable control modes</td><td>Value of the Available Closed-Loop Control Modes parameter</td></tr>
<tr><td>6, 7, 10</td><td>0x000004C0 (default setting)</td></tr>
<tr><td>7, 11</td><td>0x00000880</td></tr>
<tr><td>1, 6, 7, 8, 9, 10, 11</td><td>0x00000FC2</td></tr>
</table>
</td></tr>
</table>

INFORMATION

If the settings that were changed in the volatile memory are to be maintained when the C-413 is switched off or rebooted, they have to be saved with WPA (p. 229), refer also to "Adapting Settings" (p. 257).

INFORMATION

When the control mode is changed, the target value for the control variable is set as follows:

- The new control variable is the position: The target value is set to the current value of the position.
- The new control variable is the velocity or the force: The target value is set to zero.

INFORMATION

In the following cases, it is not permissible to change the selected control mode:

- The wave generator is running for the axis.
- An input signal channel is used as the control source for the axis.

The units of the control variables are set by PI (e.g. via the input and output matrix and linearization coefficients). Usual settings for units:

- Position: Millimeters or micrometers, depending on the travel range
- Velocity: Millimeters per second or micrometers per second, depending on the travel range
- Force: Newtons

The unit symbols for the different control variables are specified by the following parameters:

Parameter	Description and possible values
Position Axis Unit 0x07000601	Unit symbol for the position of the axis Maximum of 20 characters.
Velocity Axis Unit 0x07000603	Unit symbol for the velocity of the axis Maximum of 20 characters.
Force Axis Unit 0x07000604	Unit symbol for the force of the axis Maximum of 20 characters.

INFORMATION

The values of the parameters 0x07000601, 0x07000603 and 0x07000604 are not evaluated by the C-413 but are only used by the PC software for display purposes.

3.5.9 Generation of the Dynamics Profile

By default, the target value for closed-loop operation is input into the servo algorithm (p. 33) via the profile generator. The profile generator calculates the corresponding dynamics profile from the target value. The dynamics profile specifies the following for each point in time of the motion:

- Dynamics profile for force: Target force, first derivative of force ("jerk"), second derivative of force ("jounce")
- Dynamics profile for position: Target position, velocity, acceleration
- Dynamics profile for velocity: Target velocity, acceleration

The profile generator can be configured with the following commands:

Command	Syntax	Function
`VEL`	VEL {<AxisID> <Velocity>}	Sets the velocity for the axis. Is limited by the value of the ***Profile Generator Maximum***

Command	Syntax	Function
		Velocity parameter. The effect of the command depends on the control mode selected: ▪ The control variable is the position or the force: VEL sets the currently valid maximum velocity. ▪ The control variable is the velocity: VEL specifies the target value of the velocity. Refer to description of the VEL command (p. 215) for details.
`VEL?`	VEL? [{<AxisID>}]	The interpretation of the queried value depends on the control mode selected: ▪ The control variable is the position or the force: VEL? gets the currently valid maximum velocity. ▪ The control variable is the velocity: VEL? gets the currently valid target value of the velocity.

The profile generator can be configured with the following parameters:

Parameters	Description and possible values
Profile Generator Maximum Acceleration 0x06010000	Maximum acceleration for closed-loop operation Limits the acceleration of the dynamics profile when the control variable is the position or velocity. Is also used for deceleration.
Profile Generator Maximum Jerk 0x06010100	Maximum velocity for closed-loop operation Limits the velocity of the dynamics profile when the control variable is the force (the first derivative of force for the C-413 is designated as "jerk").
Profile Generator Enable 0x06010300	Determines the activation state of the profile generator: 0 = Profile generator deactivated 1 = Profile generator activated (default setting) In the following cases, deactivating the profile generator can improve the dynamic behavior of the axis: ▪ The wave generator is running for the axis. ▪ An analog input is used as the control source for the axis.
Profile Generator Maximum Velocity 0x06010400	Maximum velocity for closed-loop operation Limits the velocity that can be set with `VEL`. Limits the velocity of the dynamics profile when the control variable is the position. Further use when the control variable is the position or force: The velocity is set to the value of the parameter when the C-413 is switched on or rebooted and when you switch from velocity control to position or force control. Changing the parameter value in the volatile memory overwrites the velocity currently set with `VEL`.

Parameters	Description and possible values
Profile Generator Maximum Jounce 0x06010600	Maximum acceleration for closed-loop operation Limits the acceleration of the dynamics profile when the control variable is the force (the second derivative of force for the C-413 is designated as "jounce").

Dynamics profile for the position

The profile generator for the position only supports trapezoidal velocity profiles: The axis accelerates linearly (based on the maximum acceleration value) until it reaches the maximum velocity. It continues to move with this velocity until it decelerates linearly (also based on the maximum acceleration value) and stops at the specified target position.

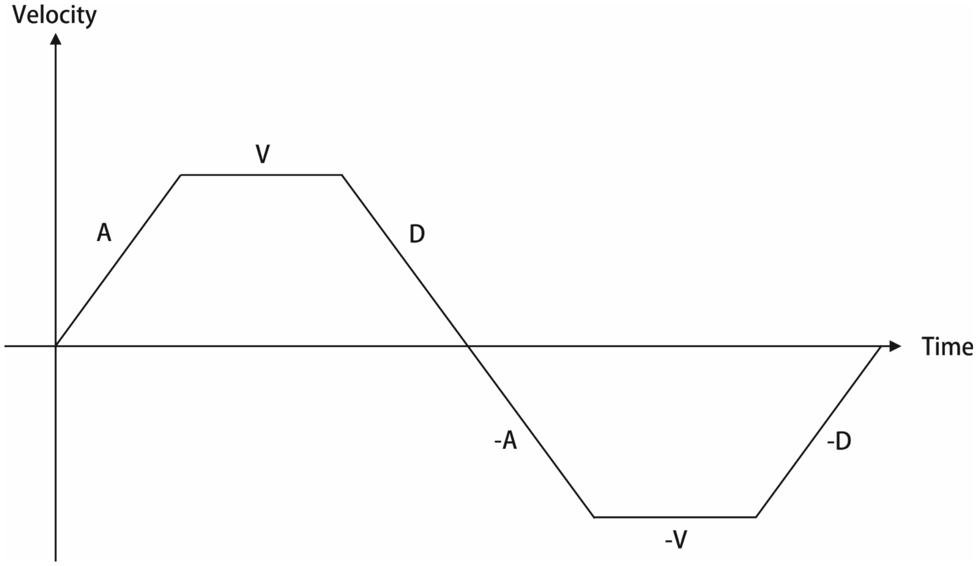


Figure 9: Basic trapezoidal velocity profile; A = acceleration, D = deceleration, V = velocity

If the deceleration has to begin before the axis reaches the maximum velocity, the profile will not have a constant velocity component and the trapezoid will become a triangle.

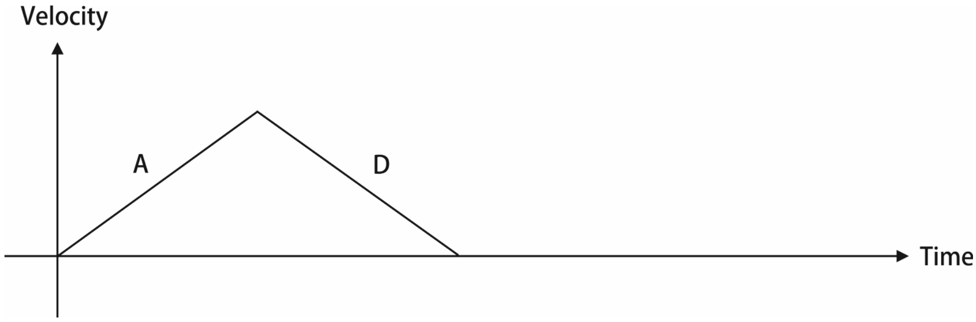


Figure 10: Basic trapezoidal velocity profile; A = acceleration, D = deceleration, no constant velocity

The target position, maximum velocity and maximum acceleration can be changed while the axis is in motion. The profile generator will always try to stay within the specified limits. If the target position is changed during the motion so that overshooting is unavoidable, the profile generator will decelerate to a complete stop and reverse the direction of motion in order to reach the specified position.

3.5.10 Servo Algorithm and Other Control Value Corrections

The settling behavior of the system can be optimized by corrections:

- Closed-loop operation: servo algorithm (PID, optional feedforward)
- Open-loop and closed-loop operation: notch filter

Control algorithms

In closed-loop operation, a profile generator calculates the dynamics profile (p. 30) from the target value (default setting; profile generator can be deactivated). The error that results from the difference between the target value or calculated dynamics profile and the current value of the control variable runs through a PID servo algorithm. Feedforward can be used optionally to improve the tracking performance and minimize the following error. The result is the control value of the axis.

When the control variable is the force (control modes with the IDs 8, 9, 10, 11), the C-413 switches between the two sets of servo control parameters for the force. Switching is intended to increase the dynamics of the axis when there is no contact between the moving part of the mechanics and a surface. Switching takes place according to the following criteria (refer to "Contact Detection in Force Control" (p. 43) for details):

- Is there contact between the moving part of the mechanics and a surface?
- Does the moving part of the mechanics work in push mode, in push/pull mode or in pull mode?

When there is no contact between the moving part of the mechanics and a surface, the C-413 also dynamically adapts the P term of the force control to the target force.

Basic structure of the PID servo algorithm for the C-413:

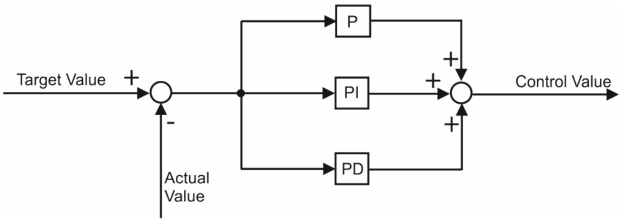


Figure 11: PID algorithm, basic structure

Based on the basic structure of the PID servo algorithm, the servo algorithms for the different control modes have the following structure.

Some of the servo algorithms have a cascade structure. In the case of a cascade structure, the "inner" control variables of the position and/or velocity are controlled in addition to the "outer" (actual) control variable. The target value for inner control variables is specified by the corresponding outer servo loop and a feedforward component.

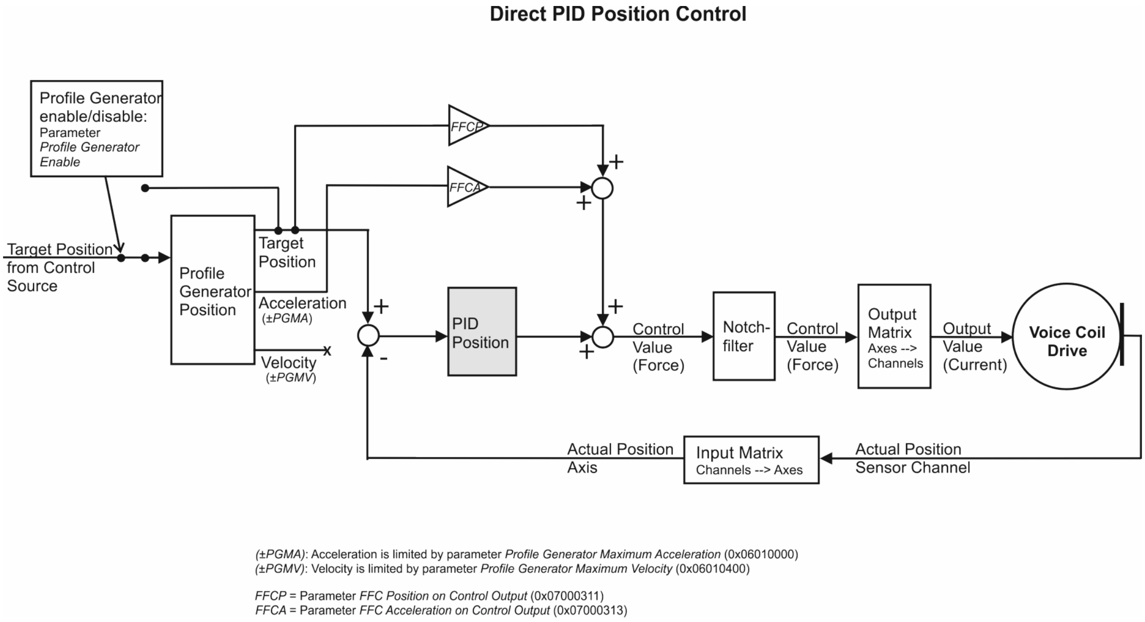


Figure 12: Direct PID position control, ID for selecting the control mode: 1

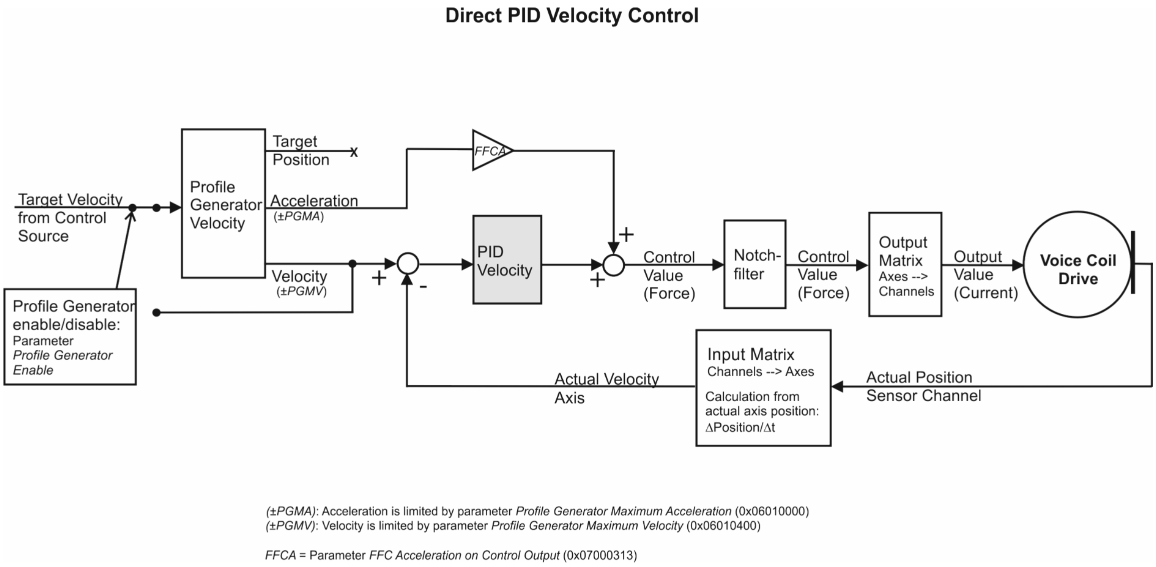


Figure 13: Direct PID velocity control, ID for selecting the control mode: 6

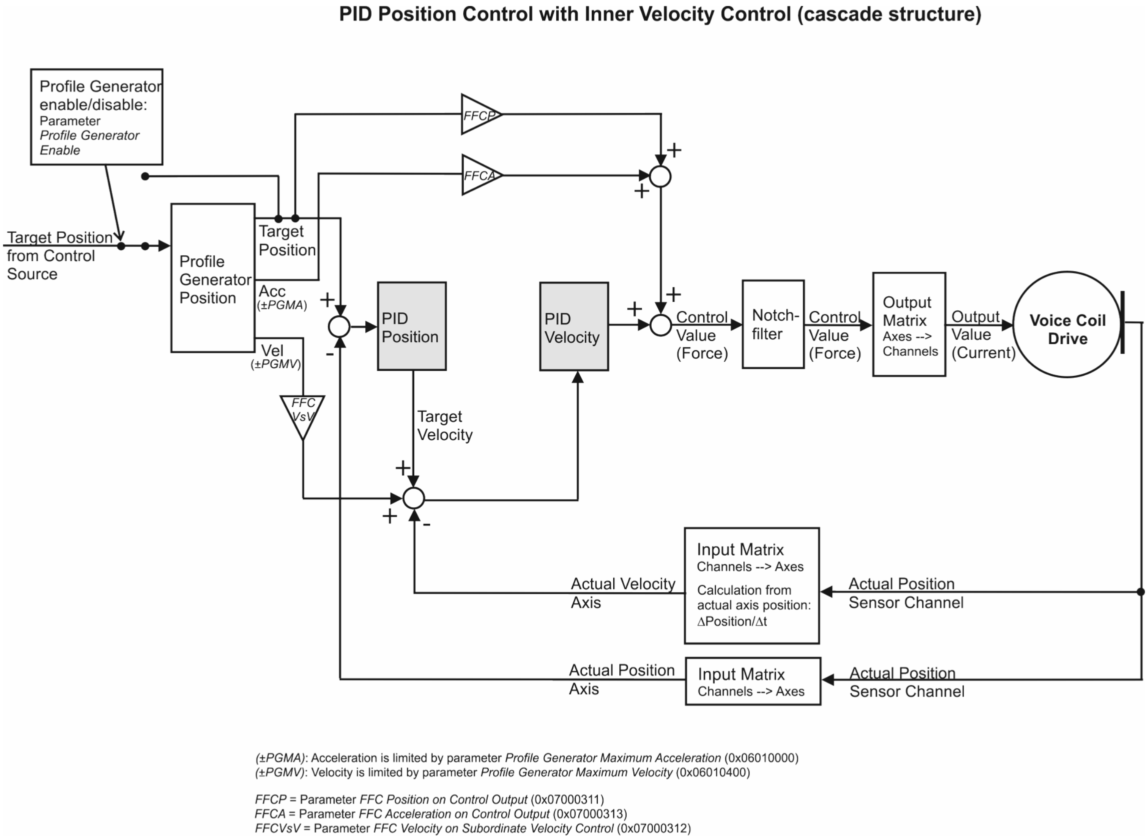
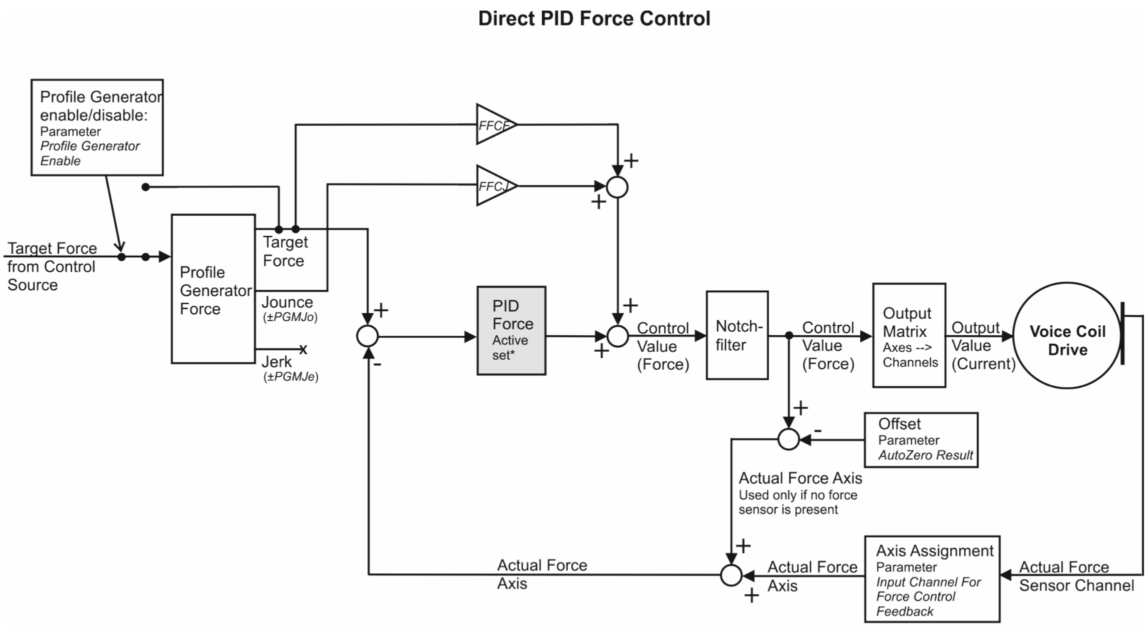


Figure 14: PID position control with velocity control, ID for selecting the control mode: 7



* Depending on the contact to the surface the C-413 switches between two sets of PID parameters for force control.

(±PGMJo): Jounce is limited by parameter *Profile Generator Maximum Jounce* (0x06010600)
(±PGMJe): Jerk is limited by parameter *Profile Generator Maximum Jerk* (0x06010100)

FFCF = Parameter *FFC Force on Control Output* (0x07000314)
FFCJ = Parameter *FFC Jounce on Control Output* (0x07000317)

Figure 15: Direct PID force control, ID for selecting the control mode: 8

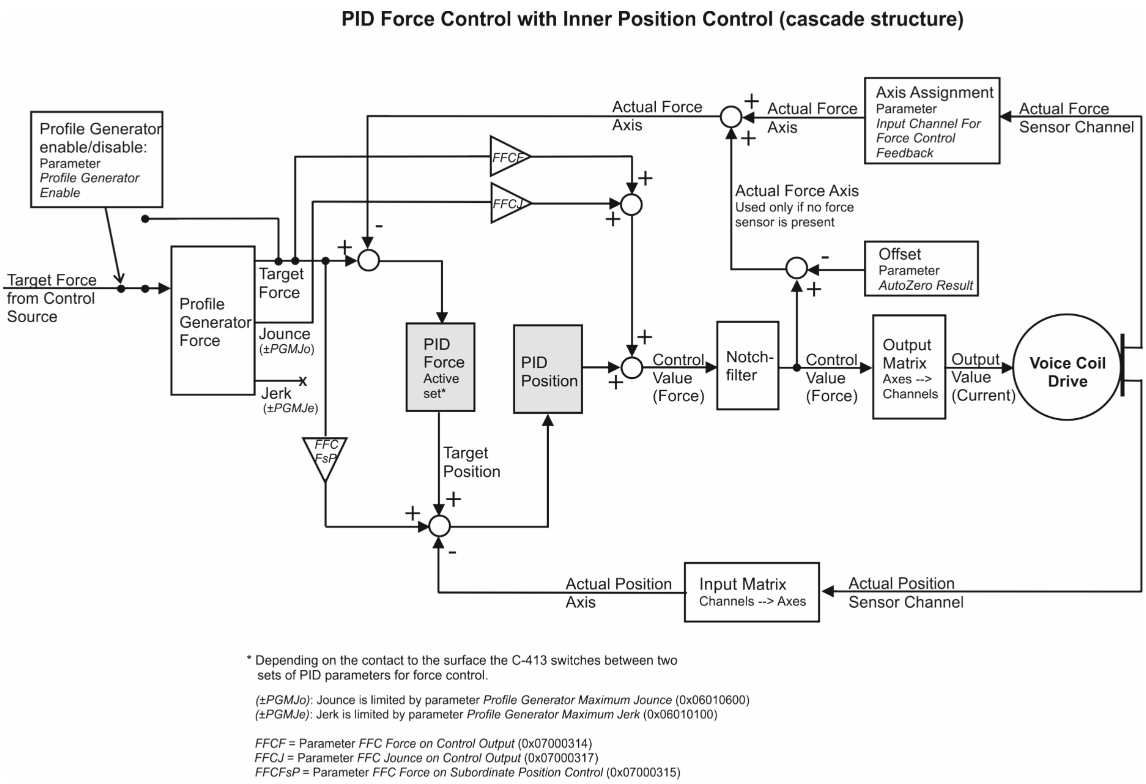


Figure 16: PID force control with position control; ID for selecting the control mode: 9

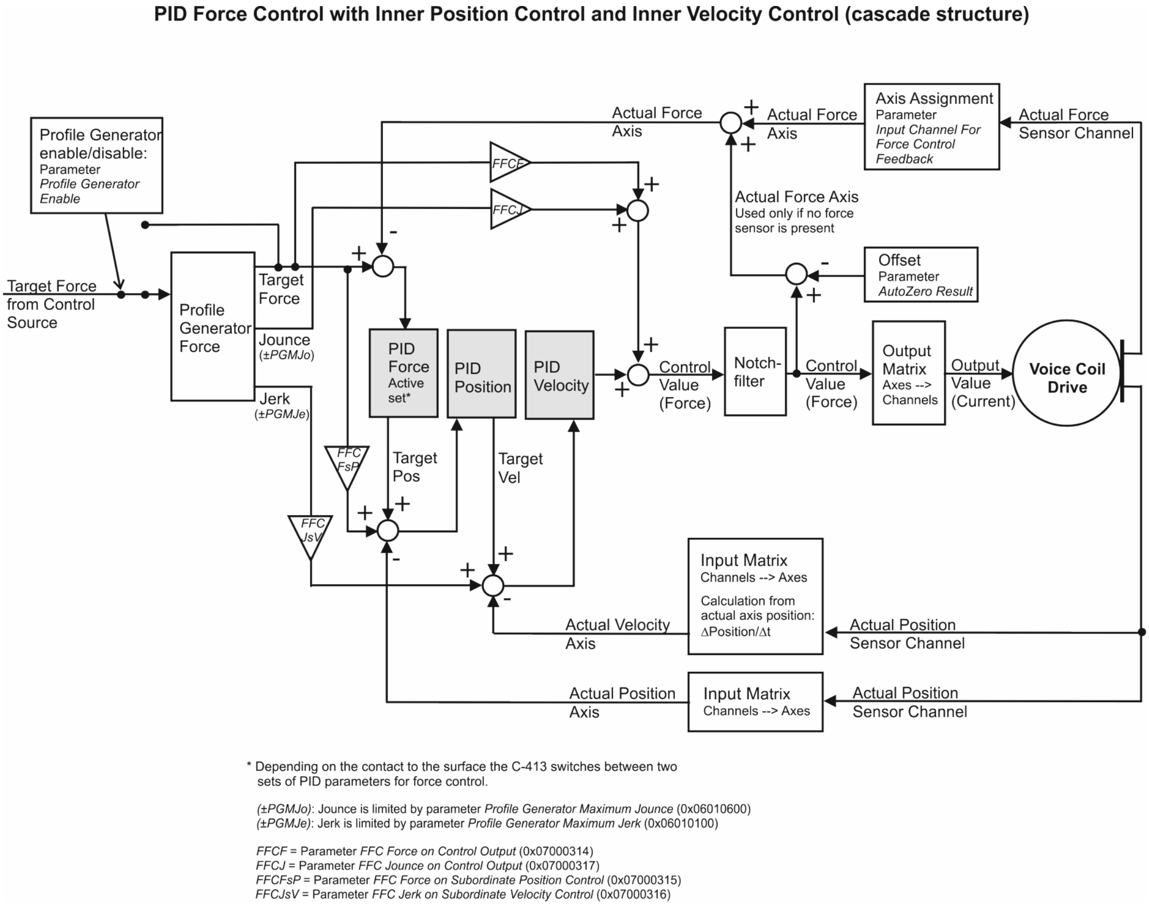


Figure 17: PID force control with position control and velocity control, ID for selecting the control mode: 10

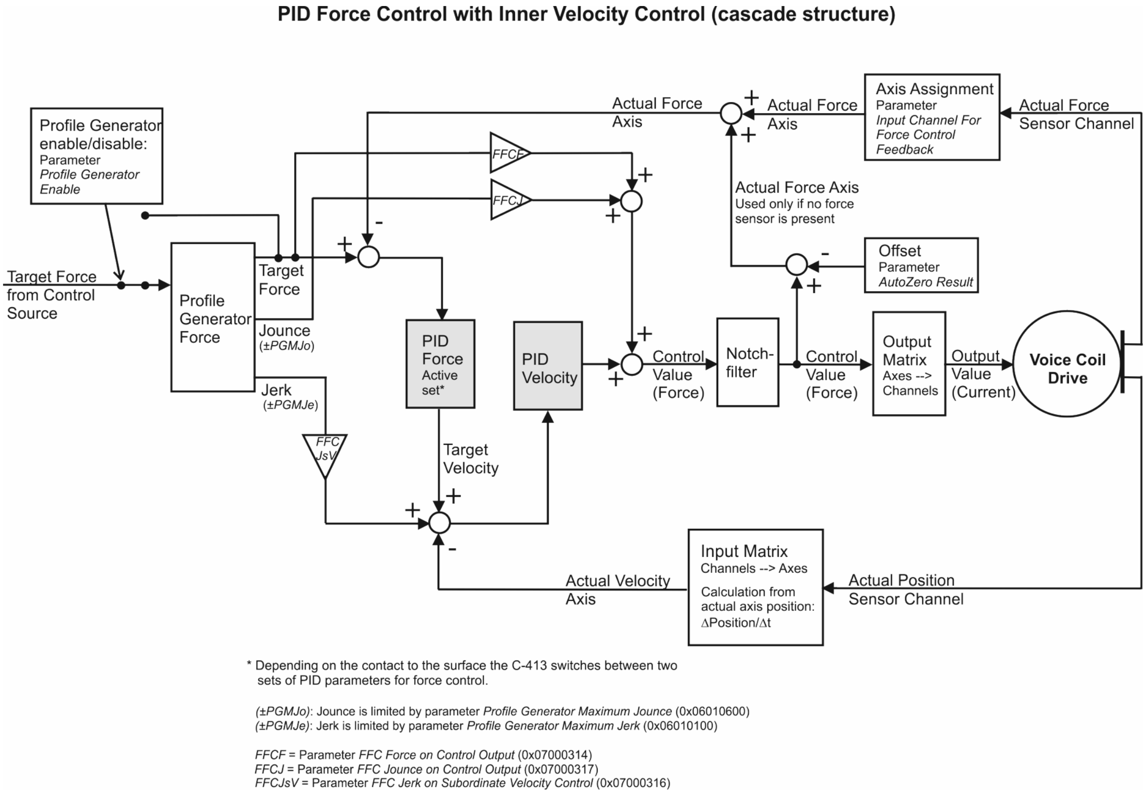


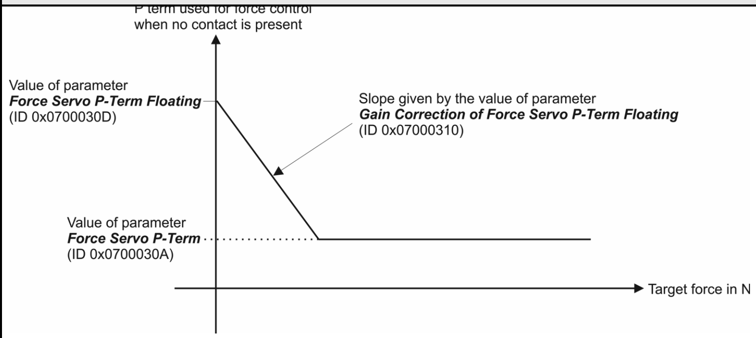
Figure 18: PID force control with velocity control, ID for selecting the control mode: 11

Parameters

The PID control algorithms use the following parameters:

Parameters	Description and Possible Values
Position Servo P-Term 0x07000300	Proportional constant (dimensionless) for the position control Integer value ≥ 0
Position Servo I-Term 0x07000301	Integration constant (dimensionless) for position control Integer value ≥ 0
Position Servo D Term 0x07000302	Differential constant (dimensionless) for position control Integer value ≥ 0
Velocity Servo P-Term 0x07000307	Proportional constant (dimensionless) for velocity control Integer value ≥ 0
Velocity Servo I-Term 0x07000308	Integration constant (dimensionless) for velocity control Integer value ≥ 0

Parameters	Description and Possible Values
Velocity Servo D-Term 0x07000309	Differential constant (dimensionless) for velocity control Integer value ≥ 0
Force Servo P-Term 0x0700030A	Proportional constant (dimensionless) for force control Integer value ≥ 0 This value is used (is "active") when there is contact between the moving part of the mechanics and a surface.
Force Servo I-Term 0x0700030B	Integration constant (dimensionless) for force control Integer value ≥ 0 This value is used (is "active") when there is contact between the moving part of the mechanics and a surface.
Force Servo D-Term 0x0700030C	Differential constant (dimensionless) for force control Integer value ≥ 0 This value is used (is "active") when there is contact between the moving part of the mechanics and a surface.
Force Servo P-Term Floating 0x0700030D	Proportional constant (dimensionless) for force control Integer value ≥ 0 If there is **no** contact between the moving part of the mechanics and a surface, this value is used (is "active") to increase the dynamics of the axis.
Force Servo I-Term Floating 0x0700030E	Integration constant (dimensionless) for force control Integer value ≥ 0 If there is **no** contact between the moving part of the mechanics and a surface, this value is used (is "active") to increase the dynamics of the axis.
Force Servo D-Term Floating 0x0700030F	Differential constant (dimensionless) for force control Integer value ≥ 0 If there is **no** contact between the moving part of the mechanics and a surface, this value is used (is "active") to increase the dynamics of the axis.
Gain Correction of Force Servo P-Term Floating 0x07000310	Correction factor for the P term of the force control (unit: 1/N) When there is **no** contact between the moving part of the mechanics and a surface, the P term is adapted dynamically to the target force by the correction factor. The value range of the P term is limited by the value of the ***Force Servo P-Term Floating*** parameter (maximum; ID 0x0700030D) and the value of the ***Force Servo P-Term*** parameter (minimum; ID 0x0700030A).

Parameters	Description and Possible Values
	

INFORMATION

The optimal values of the servo control parameters of the C-413 depend on the selected control mode and the application. In particular, the parameter values for controlling a control variable directly differ strongly from the values that are required for control with a cascade structure.

- Check the values of the servo control parameters each time that the control mode is changed.
- If necessary: Optimize the servo control parameters.

The ***Available Closed-Loop Control Modes*** (ID 0x07030101) is intended to prevent inadvertent selection of a control mode where the servo control parameters of the C-413 are not adapted; refer to "Control Modes and Control Variables" (p. 28).

- Change the value of the ***Available Closed-Loop Control Modes*** parameter only if necessary.

Feedforward can be used optionally to improve the tracking performance and minimize the following error. The target value and/or the acceleration/jounce of the dynamics profile are added to the control value of the PID servo algorithm as feedforward components. When the servo algorithm has a cascade structure, the target force and/or the velocity/jerk of the dynamics profile are also used as internal feedforward components. Refer to the figures of the servo algorithms above for details.

The feedforward components can be configured via gain values. If a gain value has the value zero, the corresponding feedforward component is deactivated (default setting for all feedforward components).

The gain values for the feedforward components can be configured via the following parameters:

Parameters	Description and Possible Values
FFC Position on Control Output 0x07000311	Gain value for the target position as feedforward component of the control value (dimensionless) Is used when the control variable is the position.

Parameters	Description and Possible Values
FFC Velocity on subordinate Velocity Control 0x07000312	Gain value for the velocity as internal feedforward component (dimensionless) Recommended value: 1.0 Is used for position control with inner velocity control (control mode 7). The velocity of the dynamics profile is multiplied by the gain value and the result is added to the output of the position control. The sum is the target value for velocity control.
FFC Acceleration on Control Output 0x07000313	Gain value for acceleration as feedforward component of the control value (dimensionless) Is used when the control variable is the position or the velocity.
FFC Force on Control Output 0x07000314	Gain value for the target force as feedforward component of the control value (dimensionless) Is used when the control variable is the force.
FFC Force on subordinate Position Control 0x07000315	Gain value for the target force as internal feedforward component (dimensionless) Is used for force control with inner position control (control modes 9 and 10). The target force is multiplied by the gain value and the result is added to the output of the force control. The sum is the target value for position control.
FFC Jerk on subordinate Velocity Control 0x07000316	Gain value for jerk as internal feedforward component (dimensionless) Is used for force control with inner velocity control (control modes 10 and 11). The jerk of the dynamics profile is multiplied by the gain value and the result is added to the output of the outer servo loop. The sum is the target value for velocity control.
FFC Jounce on Control Output 0x07000317	Gain value for the jounce as feedforward component of the control value (dimensionless) Is used when the control variable is the force.

The notch filters correct the control value both in closed-loop and open-loop operation. Parameters for adapting the notch filters:

Parameters	Description and Possible Values
Notch Frequency 0x08000100	Frequency of the first notch filter in Hz The appropriate frequency component is reduced in the control value to compensate for undesired resonances in the mechanics.
Notch Frequency 0x08000101	Frequency of the second notch filter in Hz The appropriate frequency component is reduced in the control value to compensate for undesired resonances in the mechanics.

Parameters	Description and Possible Values
Notch Rejection 0x08000200	Damping value of the first notch filter 0 to 1 Recommended value for maximum damping: 0.05. A damping value of 1 deactivates the first notch filter.
Notch Rejection 0x08000201	Damping value of the second notch filter 0 to 1 Recommended value for maximum damping: 0.05. A damping value of 1 deactivates the second notch filter.
Notch Bandwidth 0x08000300	Bandwidth of the first notch filter
Notch Bandwidth 0x08000301	Bandwidth of the second notch filter

3.5.11 Contact Detection in Force Control

When the control variable is the force, a force threshold is used to determine whether there is contact between the moving part of the mechanics and a surface.

The force threshold for contact detection can be specified as absolute or relative value. Selection takes place via the value of the ***Force Servo Surface Detection Method*** parameter (ID 0x07000406):

- 1 = The force threshold is specified as absolute value by the ***Force Sensor Surface Detection Level*** parameter (ID 0x07000401).
- 2 = The force threshold is specified as relative value on the basis for the target force; the ratio is determined by the ***Force Sensor Surface Detection Ratio*** parameter (ID 0x07000405):

  Force threshold = target force * parameter value

  The smallest possible value of the force threshold is specified by the ***Force Sensor Surface Detection Level*** parameter (ID 0x07000401).

Depending on the operating mode of the moving part, contact detection is as follows:

- If the moving part is working in push mode (***Force Control Working Mode*** parameter (ID 0x07030105) has the value 0), contact is detected in the following case:

  Current value of the force > value of the force threshold

  The result of force detection is reset to "no contact" when the current force takes a value that is smaller than the value of the force threshold for resetting (***Force Sensor Surface Lost Level*** parameter (ID 0x07000402)) for at least the duration of a delay time (***Force Sensor Surface Lost Timing*** parameter (ID 0x07000403)).
- If the moving part is working in pull mode (***Force Control Working Mode*** parameter (ID 0x07030105) has the value 2), contact is detected in the following case:

  Current value of the force < value of the force threshold (should have a negative sign)

The result of force detection is reset to "no contact" if the current force takes a value that is greater than the value of the force threshold for resetting (***Force Sensor Surface Lost Level*** parameter (ID 0x07000402)) for at least the duration of a delay time (***Force Sensor Surface Lost Timing*** parameter (ID 0x07000403)).

- When the moving part is working in push/pull mode (***Force Control Working Mode*** parameter (ID 0x07030105) has the value 1), it is assumed that there is always contact, and the force thresholds and delay time are ignored.

Depending on contact, the C-413 switches between two sets of servo control parameters for the force. Switching is intended to increase the dynamics of the axis when there is no contact. The C-413 also adapts the P term of force control to the target force dynamically. Refer to "Servo Algorithm and Other Control Value Corrections" (p. 33) for details.

Parameters

Contact detection can be configured with the following parameters:

Parameters	Description and possible values
Force Sensor Surface Detection Level 0x07000401	Absolute force threshold for the contact detection (unit: N)
Force Sensor Surface Lost Level 0x07000402	Force threshold for resetting the contact detection to "no contact" (unit: N)
Force Sensor Surface Lost Timing 0x07000403	Delay time for resetting the contact detection to "no contact", in seconds Minimum possible value: 0 s
Force Sensor Surface Detection Ratio 0x07000405	Ratio of force threshold to target force Is used to calculate the force threshold for contact detection if the ***Force Servo Surface Detection Method*** parameter (ID 0x07000406) has the value 2.
Force Servo Surface Detection Method 0x07000406	Determination of the force threshold for contact detection 1 = Force threshold is an absolute value 2 = Force threshold is a relative value based on the target force; the ratio is determined by the ***Force Sensor Surface Detection Ratio*** parameter
Force Control Working Mode 0x07030105	Working mode of the moving part of the mechanics in force control: 0 = Push mode: The moving part of the mechanics presses against a surface (normally for motion in a positive direction with positive target force). 1 = Push/pull mode: The moving part of the mechanics always works against a force. 2 = Pull mode: The moving part of the mechanics pulls on a surface (normally for motion in a negative direction with negative target force).

INFORMATION

When the control variable is the force, the target force should be set as follows to prevent overshoot:

- In push mode (***Force Control Working Mode*** parameter (ID 0x07030105) has the value 0), the target force should not be smaller than the force threshold for contact detection (***Force Sensor Surface Detection Level*** parameter (ID 0x07000401)).
- In pull mode (***Force Control Working Mode*** parameter (ID 0x07030105) has the value 2), the target force should not be greater than the force threshold for contact detection (***Force Sensor Surface Detection Level*** parameter (ID 0x07000401)).

3.5.12 On-Target State

In closed-loop operation, the on-target state can be used to check whether the target value has been reached:

- On-target state = True (1): The target value is considered to be reached
- On-target state = False (0): The target value has not been reached

The C-413 determines the on-target state based on the following criteria:

- Settling window around the target value
- Delay time for setting the on-target state.

The on-target state has the value **true** in the following cases:

- The current value of the control variable is inside the settling window and stays there at least for the duration of the delay time.
- If the value for the delay time is set to 0: The current value of the control variable is in the settling window.

The on-target state can be read with the `ONT?` and `SRG?` commands.

In the *On Target* trigger mode (p. 88) the on-target state of the selected axis is output at the selected trigger output.

Depending on the selected control variable (p. 28), the settling window and delay time are set by the following parameters:

Parameter	Description and possible values
Position On Target Tolerance 0x07000900	Settling window around the target position Specifies the window limits when the control variable is the position. If the current position exits the settling window, the target position is no longer considered as reached. The parameter value corresponds to half the width of the window.
Position On Target Settling Time 0x07000901	Delay time for setting the on-target state, in seconds Minimum possible value: 0 s Is only used when the control variable is the position.
Velocity On Target Tolerance 0x07000902	Settling window around the target velocity Specifies the window limits when the control variable is the velocity. If the current velocity exits the settling window, the target velocity is no

Parameter	Description and possible values
	longer considered to be reached. The parameter value corresponds to half the width of the window.
Velocity On Target Settling Time 0x07000903	Delay time for setting the on-target state, in seconds Minimum possible value: 0 s Is only used when the control variable is the velocity.
Force On Target Tolerance 0x07000904	Settling window around the target force Specifies the window limits when the control variable is the force. If the current force exits the settling window, the target force is no longer considered to be reached. The parameter value corresponds to half the width of the window.
Force On Target Settling Time 0x07000905	Delay time for setting the on-target state, in seconds Minimum possible value: 0 s Is only used when the control variable is the force.

INFORMATION

In control modes 7, 9, 10, and 11 (p. 28), the "inner" control variables of the position and/or velocity are controlled in addition to the "outer" (actual) control variable. The on-target status is not determined for internal control variables. The on-target status is **not** determined for inner control variables.

3.5.13 Referencing

The incremental sensors that are used for axis position feedback only return relative motion information. As a result, the controller does not know the absolute position of an axis when switching on. The axis must therefore be referenced so that absolute target positions can be commanded and reached.

Referencing can be done in different ways:

- **Referencing move** (default): A referencing move moves the axis to the reference switch, i.e., to a specifically defined point. At this point, the current position is set to a defined value (refer to the table of commands below for details). The controller now knows the absolute axis position.
- **Setting the absolute position manually**: If this referencing method is selected by the `RON` command (p. 194), you can set the current axis position to any value at any point with the `POS` command (p. 192). The axis is not moved here. The controller knows the absolute axis position afterwards.

INFORMATION

The C-413 receives the signals from the reference switches for input signal channels 1 to 4 at the **Motor & Sensor** sockets (p. 294). The reference switches are permanently allocated to input signal channels 1 to 4. The input signal channels and therefore also the reference switches are allocated to the logical axes via the input matrix (p. 16).

INFORMATION

When PIMikroMove is used for starting, a referencing move is done by default. Knowledge of the commands and parameters described here is not needed for referencing using PIMikroMove.

Commands

The following commands are available for referencing:

Command	Syntax	Function
`RON`	RON {<AxisID> <ReferenceOn>}	Selects the referencing method: ▪ <ReferenceOn> = 0: An absolute position value can be assigned with POS. Using FRF is not permitted. ▪ <ReferenceOn> = 1: A referencing move must be started with FRF. Using POS is not permitted.
`RON?`	RON? [{<AxisID>}]	Queries the referencing method.
`FRF`	FRF [{<AxisID>}]	Sequence of the referencing move: 1. The axis moves to the reference switch. 2. At the reference switch, the value of the ***Sensor Mech. Correction 1*** parameter (ID 0x02000200) is set in the volatile memory to an offset value from the ID chip of the sensor. 3. The value of the ***Sensor Mech. Correction 1*** parameter is set as the new current position of the axis. 4. The referencing move ends at the zero position of the axis. The value of the ***Sensor Mech. Correction 1*** parameter determines the behavior: – When the parameter value is zero: The axis stays at the reference switch. – When the parameter value is not zero: The axis moves from the reference switch to the new zero position. The ***Sensor Mech. Correction 1*** parameter of the input signal channel that is allocated to the axis via the input matrix (p. 16) is used .
`FRF?`	FRF? [{<AxisID>}]	Queries whether the specified axis is already referenced. 1 = axis is referenced 0 = axis not referenced
`POS`	POS {<AxisID> <Position>}	Sets the current position (does not trigger motion) and therefore references the axis.

Parameters

Referencing moves can be configured with the following parameters:

<table>
<tr><th>Parameters</th><th>Description and possible values</th></tr>
<tr><td>Sensor Mech. Correction 1
0x02000200</td><td>Offset of the polynomial for mechanics linearization.
The setting is loaded into the C-413 from the ID chip (p. 53) of the mechanics.
The value of the parameter is used during the referencing move, see the description for FRF.</td></tr>
<tr><td>Sensor Reference Signal Inversion
0x02001000</td><td>Inversion of the reference switch signal
The value of the parameter is the hexadecimal sum of the following bits:<table><tr><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr><tr><td>Negative limit switch polarity</td><td>Positive limit switch polarity</td><td>Reference signal polarity</td><td>Reference signal edge</td></tr></table>0 (bit not set) = Signal not inverted
1 (bit set) = Signal inverted
Because the C-413 does not have any inputs for limit switches, bits 2 and 3 are not evaluated (set to 1 by default) Bits 0 and 1 are set to 0 by default. The corresponding parameter value is 0x0000000C.
Example for inversion of the reference switch signal:
When the parameter value is 0x00000003, the reference switch signal is inverted and the referencing move refers to the falling edge of the reference switch signal. The limit switch signals are not inverted (setting is irrelevant).</td></tr>
<tr><td>Velocity For Reference Move
0x07030300</td><td>Velocity for referencing moves
Specifies the maximum velocity for approaching the reference switch during a referencing move. For high repeatability during referencing, the maximum of this value should be as large as the value of parameter 0x06010400. If the value of parameter 0x07030300 is set to 0, referencing moves are not possible.</td></tr>
<tr><td>Power Up Reference Move Enable
0x07000806</td><td>Automatic execution of the referencing move after the C-413 is switched on or rebooted
0 = Referencing move is not automatically executed
1 = Referencing move is automatically executed</td></tr>
</table>

INFORMATION

If the axis has not yet been referenced, relative motion is possible in closed-loop operation with CTR (p. 168), MVR (p. 189), STE (p. 202), and IMP (p. 187) (depending on the current selected referencing method).

INFORMATION

If the absolute position of the axis is defined manually with the POS command, conflicts with the settings for the travel range limits can occur (parameter 0x07000001, query with `TMX?`, and 0x07000000, query with `TMN?`).

➢ Set the absolute position of the axis manually only if referencing is not otherwise possible.

3.5.14 Autozero Procedure for Compensating the Weight Force

A voice coil drive does not have any self-locking. Switching off the servo mode for the axis can therefore lead to unexpected motions. Typical case: In the case of a vertically oriented motion axis, the moving part falls down due to its weight force when the servo mode is switched off.

After a successful autozero procedure, the weight force of the moving part can be compensated for when the servo mode is switched off: The autozero procedure defines the control value (p. 24) at which the axis generates a force of 0 N in open-loop operation and thereby maintains the current position.

When a force sensor is present for the axis, i.e., when the force sensor is directly allocated to the axis via the ***Input Channel For Force Feedback*** parameter (ID 0x07000400) (p. 19), the autozero procedure also adapts the sensor value; see below for details.

INFORMATION

When the C-413 is switched off or rebooted, the weight force of the moving part can no longer be compensated for even with a successful autozero procedure.

➢ Before switching off or rebooting the C-413, take suitable measures to ensure that no unexpected motions are possible.

INFORMATION

Knowledge of the commands and parameters described here is not needed when the autozero procedure is executed with PIMikroMove

Commands

The following commands are available for the autozero procedure:

Command	Syntax	Function
`ATZ`	ATZ [{<AxisID> <LowValue>}]	Starts an automatic zero point adjustment ("autozero procedure") in which the axis is moved. <LowValue> specifies the position at which to determine the control value that is required to generate a force of 0 N in open-loop operation. When "NaN" is entered for <LowValue>, the value of the ***AutoZero Low Value*** parameter (ID 0x07000A00) is used. The adjustment procedure changes the value of the ***AutoZero Result*** parameter (ID 0x07000A03) in the volatile memory.

Command	Syntax	Function
		When a force sensor is allocated to the axis via the ***Input Channel For Force Feedback*** parameter (ID 0x07000400), the adjustment procedure also changes the value of the ***Sensor Mech. Correction 1*** parameter (ID 0x02000200) for the input signal channel of the force sensor. The parameter value is changed in the volatile memory. See description of the ATZ command (p. 154) for further details, .
ATZ?	ATZ? [{<AxisID>}]	Queries the success for the automatic zero point adjustment.

Parameters

The autozero procedure can be configured with the following parameters:

Parameter	Description and Possible Values
Sensor Mech. Correction 1 0x02000200	Offset coefficient of the polynomial for mechanics linearization (p. 21) When a force sensor is allocated to an axis via the ***Input Channel For Force Feedback*** parameter (ID 0x07000400), the autozero procedure of this axis changes the value of the offset coefficient for the input signal channel of the force sensor, see also the ***Force Sensor AutoZero Value*** parameter (ID 0x07000404). The parameter value is changed in the volatile memory.
Force Sensor AutoZero Value 0x07000404	Value of the force sensor after the autozero procedure Is set by the autozero procedure if the input signal channel of the force sensor is allocated to the axis via the ***Input Channel For Force Feedback*** parameter (ID 0x07000400). In order to set the sensor value, the autozero procedure changes the value of the ***Sensor Mech. Correction 1*** parameter (ID 0x02000200) for the input signal channel.
Power Up AutoZero Enable 0x07000802	Automatic execution of the autozero procedure after the C-413 is switched on or rebooted 0 = autozero procedure is not automatically executed 1 = autozero procedure is automatically executed
AutoZero Low Value 0x07000A00	Lower limit of the motion range during the autozero procedure Is used when the value "NaN" is entered for <LowValue> in the ATZ command. Specifies the position at which the control value is to be determined that is required to generate a force of 0 N in open-loop operation. This position is then also the ending position of the autozero procedure. When the parameter value is smaller than the minimum commandable position (***Position Range Limit min*** parameter, ID 0x07000000), the minimum commandable position is used during the autozero procedure.

Parameter	Description and Possible Values
AutoZero High Value 0x07000A01	Upper limit of the motion range during the autozero procedure When the parameter value is greater than the maximum commandable position (***Position Range Limit max*** parameter, ID 0x07000001), the maximum commandable position is used during the autozero procedure.
AutoZero Result 0x07000A03	Result of the autozero procedure The autozero procedure changes the value of the parameter in the volatile memory. This value is interpreted as the force in N and is set as the control value of the axis when the servo mode is switched off.

INFORMATION

The axis must be referenced before the autozero procedure (referencing move or manual setting of the current position). Refer to "Referencing" (p. 46).

INFORMATION

Depending on the setting of the used parameters, the motion can extend over the entire travel range of the axis during the autozero procedure.

INFORMATION

If the settings that were changed in the volatile memory are to be maintained when the C-413 is switched off or rebooted, they have to be saved with WPA (p. 229), refer also to "Adapting Settings" (p. 257).

3.5.15 I2t Monitoring for Protecting the Mechanics

The C-413's I^2t monitoring can prevent the voice coil drive from overheating as a result of overcurrent. On delivery, I^2t monitoring is deactivated by default. You can activate I^2t monitoring via parameters and adapt it to the drive of your mechanics.

When I^2t monitoring is activated, the C-413 calculates the overcurrent limit I^2t_{max} from drive-specific parameters. The C-413 reduces the output current to the drive's nominal current when the current I^2t value reaches the overcurrent limit I^2t_{max}. You will find a formula-based description of the functionality underneath the parameter table.

You can record the current I^2t value with the C-413's data recorder, record option 33 (I2T Value).

Limiting the current by I^2t monitoring can be noticeable in the behavior of the mechanics, e.g., by reduced velocity or force.

Parameters

I^2t monitoring by the C-413 can be configured with the following parameters:

Parameters	Description and Possible Values
I2T Peak Current [A] 0x0C001000	Peak current I_p of the drive (unit: A) Refer to the datasheet for the mechanics. Used by the C-413 for calculating the overcurrent limit I^2t_{max}.
I2T Peak Current Time [s] 0x0C001001	Maximum duration t_p of peak current (unit: s) Refer to the datasheet for the mechanics. Used by the C-413 for calculating the overcurrent limit I^2t_{max}.
I2T Nominal Current [A] 0x0C001002	Nominal current I_n of the drive (unit: A) Refer to the datasheet for the mechanics. Used by the C-413 for calculating the current I^2t value and overcurrent limit I^2t_{max}. The C-413 limits the output current to the value of this parameter when the current I^2t value reaches the overcurrent limit I^2t_{max}.
I2T Active 0x0C001003	Determines the activation state of I^2t monitoring: 0 = I^2t monitoring is deactivated (default) 1 = I^2t monitoring is activated

Formula-based description of I^2t monitoring:

$$I^2t = \int (I^2 - {I_n}^2)\, dt$$

$$I^2t_{max} = \left({I_p}^2 - {I_n}^2\right) * t_p$$

$$I^2t \geq I^2t_{max} \rightarrow I = I_n$$

where

I = current output current of the C-413

I_n = nominal current of the drive

I_p = peak current of the drive

t_p = maximum duration of peak current

INFORMATION

If the settings that were changed in the volatile memory are to be maintained when the C-413 is switched off or rebooted, they have to be saved with WPA (p. 229), refer also to "Adapting Settings" (p. 257).

3.5.16 Deactivation of Axes

If an axis may not be moved under any circumstances, it can be deactivated. A deactivated axis is **not** accessible for axis-related commands (e.g., motion commands or position queries). The identifier of a deactivated axis **cannot** be queried.

The criterion for deactivating the axes of the C-413 is the value of the ***Number Of System Axes*** parameter (ID 0x0E000B02):

Value of the parameter	Available/deactivated axes
2	All axes (1 and 2) are available (default setting)
1	Axis 2 is deactivated, axis 1 is available
0	All axes (1 and 2) are deactivated

To deactivate/activate axes, it is necessary to change the ***Number Of System Axes*** parameter in the nonvolatile memory and then to reboot the C-413 (Refer to "Adapting Settings" (p. 257) for further information).

Deactivating an axis does not have any effect on the availability of the input signal channels and output signal channels that are allocated to this axis via matrices (p. 16).

3.5.17 ID Chip Detection

The mechanics with voice coil drives offered by PI contain one ID chip per sensor where the following data is saved:

- Information on the mechanics (available as parameters in the C-413):
  - Type (***Stage Type*** parameter, ID 0x0f000100)
  - Serial number (***Stage Serial Number*** parameter, ID 0x0f000200)
- Coefficients of the polynomial for mechanics linearization (available in the C-413 as ***Sensor Mech. Correction 1*** parameter to ***Sensor Mech. Correction 5*** parameter, IDs 0x02000200 to 0x02000600)
- Settings for the sensor: Interpolation rate, corrections to hysteresis, phase and offset, gain values, offset value for the zero position (used for referencing moves (p. 46))
- Data for checking the validity when the contents of the ID chip are read out

The data of the mechanics is loaded into the memory of the C-413 from the ID chip as follows:

- When the C-413 is switched on or rebooted, the firmware reads the type and the serial number from the ID chip.
- The data read is compared with the data stored in the C-413:
  - The type and the serial number are identical in the ID chip and the C-413: The C-413 does not load any other data from the ID chip.
  - The type or serial number are different in the ID chip and C-413: The C-413 loads all data from the ID chip.

  The data read from the ID chip is written to the volatile memory and the nonvolatile memory of the C-413.

The data that is available as parameters in the memory of the C-413 can be queried, refer to "Adapting Settings" (p. 257).

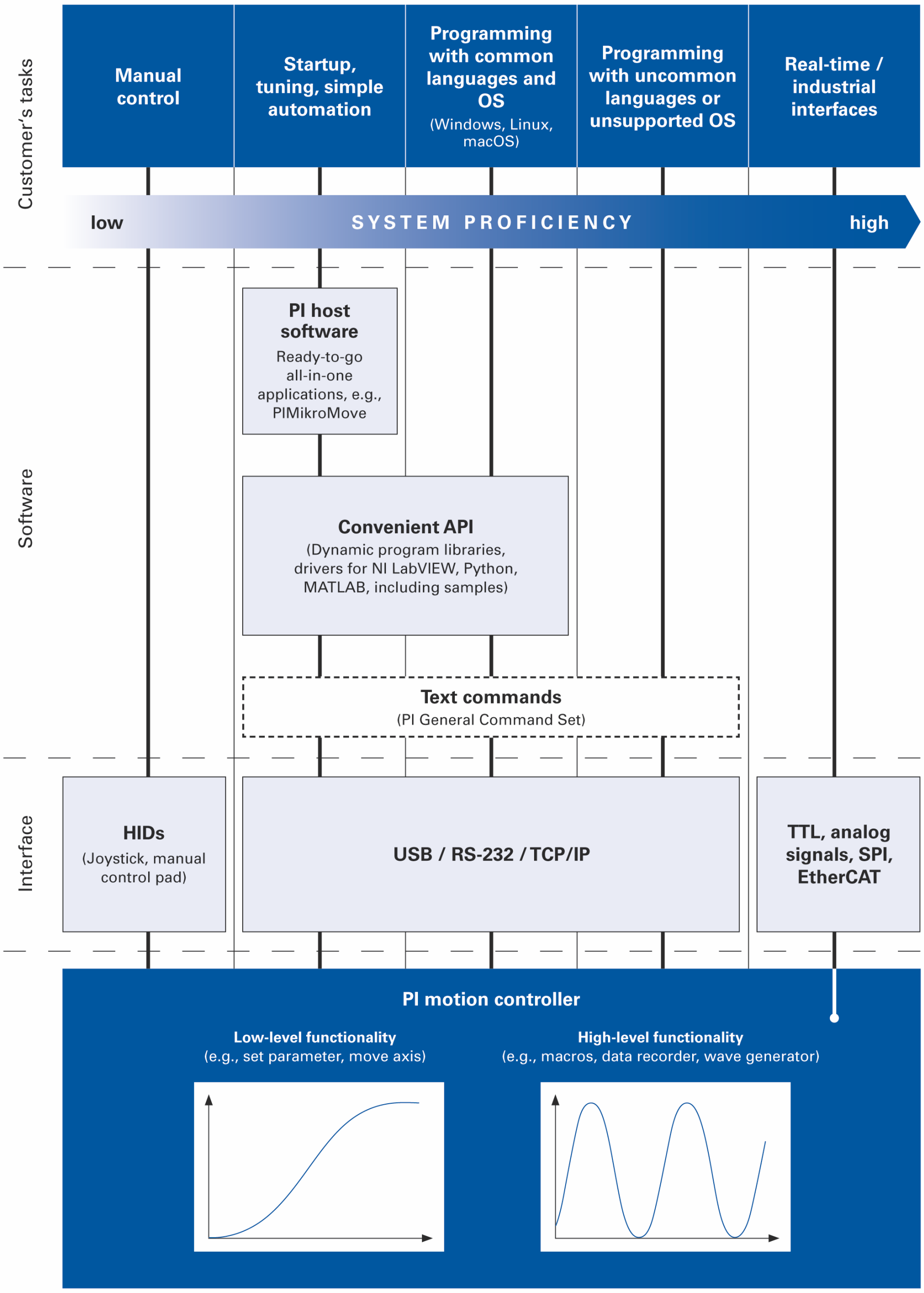
Parameters that are loaded from the ID chip are marked in color in the parameter overview (p. 267).

The data of the position sensor channels that is loaded from the ID chip is converted to axis-related information using the input matrix (p. 16) (e.g., when the type is queried with the CST? command (p. 162) or during referencing moves (p. 46)).

3.6 Communication Interfaces

3.6.1 Control of PI Systems

Basically, systems from PI can be controlled as follows:



The C-413 can be controlled from a PC with the ASCII commands of the PI General Command Set (p. 143). The connection to the PC is made via a USB connection.

In addition, the C-413 can also be controlled from an SPI master; refer to the C413T0014 Technical Note for details.

INFORMATION

In the C-413, a USB UART module is used for the USB interface. This results in the following:

- A baud rate setting is necessary for the USB interface.
- If the controller is connected via the USB connection and switched on, the USB interface is also shown as a virtual COM port in the PC software.

To successfully establish communication via USB, the baud rates of the PC and C-413 must be identical. The PIMikroMove PC software, which PI recommends for initial startup of the C-413, therefore automatically adapts the baud rate of the PC to the current baud rate of the C-413.

Only when the PC software offers the selection of the baud rate of the PC when communication is established via USB:

➢ Adapt the baud rate of the PC to the current baud rate of the C-413.

The baud rate of the C-413 can be set with the following parameter:

Parameters	Description and Possible Values
UART Baudrate 0x11000400	Baud rate for the UART of the USB interface Possible values: 9600, 14400, 19200, 38400, 57600, 115200 baud

3.7 Overview of PC Software

3.7.1 PI Software Suite

A data storage device with the PI Software Suite is included in the C-413's scope of delivery (p. 11). Some components of the PI Software Suite are described in the table below. For information on the compatibility of the software with PC operating systems see the C-990.CD1 Release News in the root directory of the data storage device.

Libraries, drivers

PC software	Short description	Recommended use
Dynamic program library for GCS	Allows software programming for the C-413 with programming languages such as C++. The functions in the dynamic program library are based on the PI General Command Set (GCS).	For users who would like to use a dynamic program library for their application. Is required for PIMikroMove. Is required for NI LabVIEW drivers.
Drivers for use with NI LabVIEW software	NI LabVIEW is a software for data acquisition and process control (must be ordered separately from National Instruments).The driver library is a collection of virtual instrument drivers for PI controllers.	For users who want to use NI LabVIEW to program their application.

PC software	Short description	Recommended use
	The drivers support the PI GCS.	
MATLAB drivers	MATLAB is a development environment and programming language for numerical calculations (must be ordered separately from MathWorks). The PI MATLAB driver consists of a MATLAB class that can be included in any MATLAB script. This class supports the PI GCS. The PI MATLAB driver does not require any additional MATLAB toolboxes.	For users who want to use MATLAB to program their application.
USB driver	Driver for the USB interface	For users who want to connect the controller to the PC via the USB interface.

User software

PC software	Short description	Recommended use
PIMikroMove	Graphic user interface for Windows with which the C-413 and other controllers from PI can be used. ▪ The system can be started without programming effort ▪ Graph of motions in open-loop and closed-loop operation ▪ Macro functionality for storing command sequences on the PC (host macros) ▪ Support of HID devices ▪ Complete environment for command entry, for trying out different commands PIMikroMove uses the dynamic program library to supply commands to the controller.	For users who want to do simple automation tasks or test their equipment before or instead of programming an application. A log window showing the commands sent makes it possible to learn how to use the commands.
PITerminal	Terminal program that can be used for nearly all PI controllers.	For users who want to send GCS commands directly to the controller.
PI Update Finder	Checks the PI software installed on the PC. If more current versions of the PC software are available on the PI server, downloading is offered.	For users who want to update the PC software.
PIFirmwareManager	Program for user support when updating firmware of the C-413.	For users who want to update the firmware.

4 Unpacking

For the C-413.20A and .20 OEM boards, the following applies:

NOTICE

Electrostatic hazard

The C-413 contains electrostatic-sensitive components (ESD) and can be damaged if handled improperly.

- Avoid touching assemblies, pins and PCB traces.
- Before you touch the C-413, discharge your body appropriately, e.g., by using an antistatic wrist strap.
- Only handle and store the C-413 in environments that dissipate existing static charges to earth in a controlled way and prevent electrostatic charges (ESD workplace or electrostatic discharge protected area, abbreviated to EPA).

1. Unpack the C-413 with care.
2. Compare the contents with the scope of delivery according to the contract and the delivery note.
3. Inspect the contents for signs of damage. If any parts are damaged or missing, contact our customer service department immediately (p. 287).
4. Keep all packaging materials in case the product needs to be returned.

5 Installing

5.1 General Notes on Installation

- Install the C-413 near the power source so that the power plug can be quickly and easily disconnected from the mains.
- Only use cables and connectors that meet local safety regulations.

5.2 Installing the PC Software

Communication between the C-413 and a PC is required to configure the C-413 and to command motion using the GCS commands. Various PC software applications are available for this purpose.

5.2.1 Doing Initial Installation

Accessories

- PC with Windows or Linux operating system and at least 30 MB free storage space
- Data storage device with PI Software Suite (included in the scope of delivery)

  For information on the compatibility of the software with PC operating systems see the C-990.CD1 Release News in the root directory of the data storage device.

Installing the PC software in Windows

1. Start the installation wizard by opening ***PISoftwareSuite.exe*** in the installation directory (root directory of the data storage device).

   The ***InstallShield Wizard*** window opens for installing the PI Software Suite.
2. Follow the instructions on the screen.

   The PI Software Suite includes the following components:
   - Drivers for use with NI LabVIEW software
   - Dynamic program library for GCS
   - PIMikroMove
   - PC software for updating the firmware of the C-413
   - PIUpdateFinder for updating the PI Software Suite
   - USB driver

INFORMATION

PIMikroMove requires NI LabVIEW Run-Time Engine to provide the ***Device Parameter Configuration*** window. The setup program therefore prompts you to start the installation assistant for NI LabVIEW Run-Time Engine after the PI software suite has been installed ("Launch NI LabWindows-CVI-RTE 2010 SP1 Installer" checkbox).

Installing the PC software on Linux

1. Unpack the tar archive from the /Linux directory of the data storage device to a directory on your PC.
2. Open a terminal and go to the directory to which you have unpacked the tar archive.
3. Log in as a superuser (root privileges).
4. To start the installation, enter ./INSTALL
   Pay attention to capitalization while entering the command.
5. Follow the instructions on the screen.

You can select individual components for installation.

5.2.2 Installing Updates

PI is constantly improving the PI Software Suite.

➢ Always install the latest version of the PI Software Suite.

Requirements

✓ Active connection to the Internet

✓ If your PC uses a Windows operating system:

- You have downloaded the PIUpdateFinder manual (A000T0028) from the PI website. The link is in the "A000T0081-Downloading Manuals from PI.pdf" file in the \Manuals folder on the data storage device with the PI Software Suite.

Updating the PC software on Windows

➢ Use the PIUpdateFinder:

- Follow the instructions in the manual for the PIUpdateFinder (A000T0028).

Updating the PC software on Linux

1. Open the website https://www.physikinstrumente.com/en/products/software-suite (https://www.physikinstrumente.com/en/products/software-suite).
2. Scroll down to ***Downloads***.
3. For ***PI Software Suite C-990.CD1***: Select ***ADD TO LIST+***
4. Select ***REQUEST***
5. Fill out the download request form and send the request.

   The download link will be sent to the email address entered in the form.
6. Unpack the archive file on your PC to a separate installation directory.
7. In the directory with the unpacked files, go to the ***linux*** subdirectory.

8. Unpack the archive file in the ***linux*** directory by entering the command tar -xvpf <name of the archive file> on the console.
9. Log into the PC as superuser (root privileges).
10. Install the update.

INFORMATION

If software is missing in the ***Downloads*** area or problems occur with downloading:

➢ Contact PI's customer service (p. 287).

5.3 Ensuring Ventilation

High temperatures can overheat the C-413.

➢ Set up the C-413 with a distance of at least 10 cm to the top and rear sides and at least 5 cm to the sides. If this is not possible, make sure that the area is cooled sufficiently.

➢ Ensure sufficient ventilation at the place of installation.

➢ Keep the ambient temperature to a non-critical level (<50° C).

5.4 Mounting the C-413

The C-413.2G and .2GA models can be used as a bench-top device or mounted in any orientation on a surface.

Tools and accessories

- Suitable screws
- Suitable screwdriver

Mounting the C-413

1. Make the necessary holes in the surface.

   The arrangement of the recesses in the mounting rails of the C-413.2G and .2GA models can be found in the dimensional drawing in "Dimensions" (p. 292).
2. Mount the C-413 in the recesses in the mounting rails with two suitable screws on each side.

5.5 Installing the C-413 into a Housing

The C-413.20A and .20 models must be installed in a suitable housing before startup.

NOTICE

Electrostatic hazard

The C-413 contains electrostatic-sensitive components (ESD) and can be damaged if handled improperly.

- ➢ Avoid touching assemblies, pins and PCB traces.
- ➢ Before you touch the C-413, discharge your body appropriately, e.g., by using an antistatic wrist strap.
- ➢ Only handle and store the C-413 in environments that dissipate existing static charges to earth in a controlled way and prevent electrostatic charges (ESD workplace or electrostatic discharge protected area, abbreviated to EPA).

INFORMATION

The C-413 is intended to be screwed into a housing. For this purpose, the board has four mounting holes with Ø 2.8 mm. The mounting holes have GND potential.

Tools and accessories

- Suitable housing:
  - The housing is shielded and designed in such a way that the C-413 fulfills all requirements for electromagnetic compatibility after installation.
  - The housing has suitable retainers for screwing in the C-413. The arrangement of the mounting holes of the C-413.20 and .20A models can be found in the dimensional drawing in "Dimensions" (p. 292).
  - The housing is connected to a suitable protective earth conductor. The retainers for screwing in the C-413 have a conductive connection to the protective earth conductor.
- Four sufficiently conductive M2.5 screws with a suitable length

Installing the C-413 into the housing

1. Mount the C-413.20 or .20A with four screws in the housing.
2. Close the housing.

5.6 Connecting the C-413 to the Protective Earth Conductor

C-413.2GA and .2G models

Figure 19: Protective earth connection of the C-413.2GA and .2G models

- ➢ Connect the threaded pin with the protective earth conductor symbol (see figure) on the housing of the C-413 to the protective earth conductor.

C-413.20A and .20 models

- ➢ Follow the instructions in "Installing the C-413 into a Housing" (p. 61).

5.7 Connecting the Power Adapter to the C-413

Requirements

- ✓ The power cord is **not** connected to the power socket.

Tools and accessories

- 24 V wide range input power supply included (for line voltages between 100 and 240 VAC at 50 or 60 Hz)
- Alternative: Sufficiently rated power supply
- Adapter for the power supply connector included; barrel connector to M8 4-pin connector (f)
- Alternative: Sufficiently sized adapter
- Power cord included
- Alternative: Sufficiently sized power cord

Connecting the power supply to the C-413

- ➢ Connect the adapter's M8 connector (f) to the C-413's 24 V connection (M8 panel plug).
- ➢ Connect the adapter's barrel connector to the power supply's barrel connector socket.
- ➢ Connect the power cord to the power supply.

5.8 Connecting the Mechanics

INFORMATION

If unsuitable cables are used, interference can occur in the signal transmission between the mechanics and the C-413.

- ➢ Only use original PI parts to connect the C-413 to the mechanics. The maximum cable length is **1 m**.
- ➢ If you need longer cables, contact our customer service department (p. 287).

INFORMATION

It is possible to assign the input signal channels and output signal channels in the C-413's firmware to the logical axes (p. 16). The allocation determines the identifiers that are to be used for commanding the connected mechanics. Allocation with the default settings of the C-413:

- Input signal channel 1 and output signal channel 1 are allocated to axis 1. Both channels are on the **Motor & Sensor 1** socket. The mechanics connected to the **Motor & Sensor 1** socket is therefore commanded as axis 1.
- Input signal channel 3 and output signal channel 2 are allocated to axis 2. Both channels are on the **Motor & Sensor 2** socket. The mechanics connected to the **Motor & Sensor 2** socket is therefore commanded as axis 2.

INFORMATION

Force sensors can be connected to the C-413 via the following connectors:

- **Motor & Sensor** sockets: Incremental force sensors with data transmission via sensor SPI. Mechanics that have a separate D-sub 15 (m) connector for the force sensor in addition to the motor connector (e.g., V-273.431 model from PI) occupies both **Motor & Sensor** sockets.
- **I/O** panel plug: Force sensors with analog output signal, refer to "Connecting Analog Signal Sources" (p. 67) for details

When a force sensor is connected and to be used, the corresponding input signal channel must be allocated to the axis via the ***Input Channel for Force Feedback*** parameter (ID 0x07000400) (p. 19).

INFORMATION

C-413 and mechanics are delivered as a preconfigured system.

- ➢ If an assignment of the connections is specified by the labeling of the C-413 and/or mechanics, adhere to this assignment when connecting the mechanics.

INFORMATION

The C-413 does the following when switching on or rebooting:

- Initializing the sensor electronics in the mechanics

- Reading the ID chips of the sensors; refer to "Detecting the ID Chip" (p. 53) for details

➢ When you have connected mechanics to the switched on C-413: Switch the C-413 off and on again, or reboot the C-413 with the `RBT` command (p. 194) or with the corresponding function of the PC software.

Requirements

✓ The C-413 is switched off, i.e., the power adapter is **not** connected to the power socket with the power cord.

Tools and accessories

- Mechanics that the C-413 is configured with at PI
- Alternative: Mechanics of the same type

Connecting the mechanics

➢ Connect the motor connector of the mechanics to a **Motor & Sensor** socket on the C-413.

➢ If the mechanics has a force sensor with a separate D-sub 15 (m) connector: Connect the force sensor to the unused **Motor & Sensor** socket on the C-413.

5.9 Connecting the PC

5.9.1 Connecting the C-413 via the USB interface

Requirements

✓ The PC has an unused USB interface.

Tools and accessories

- USB cable (type A to mini B) for connecting to the PC, included in the scope of delivery (p. 11)

Connecting the C-413 to the PC

➢ Connect the USB cable to the USB socket of the C-413 and the USB interface of the PC.

5.10 Connecting the Digital Outputs

INFORMATION

The digital outputs on the **I/O** panel plug of the C-413 can be used to trigger external devices, for example.

- Outputs 1 to 5 (pins 15, 16, 17, 18, 19): Programmable output, refer to "Digital Output Signals" (p. 85)
- Output 6 (pin 20): Output of the servo cycles, not programmable

Tools and accessories

- Suitable cable, e.g., C-413.1IO I/O cable with open end (p. 296), available as an optional accessory (p. 11)
- Suitable device with digital input, see the pin assignment of the **I/O** panel plug (p. 295) for details

Connecting a device

➢ Connect an appropriate device to one of pins 15, 16, 17, 18, 19 or 20 of the **I/O** panel plug of the C-413.

5.11 Connecting the Digital Inputs

INFORMATION

The digital inputs on pins 9, 10, 11, 12 of the **I/O** panel plug can be used to start the data recording and wave generator output of the C-413; refer to "Digital Input Signals" (p. 90).

Tools and accessories

- Suitable signal source, see the pin assignment of the **I/O** panel plug (p. 295) for details
- If necessary: Suitable cable, e.g., C-413.1IO I/O cable with open end (p. 296), available as an optional accessory (p. 11).

Connecting a digital signal source

➢ Connect an appropriate signal source to one of pins 9, 10, 11 or 12 of the **I/O** panel plug of the C-413.

5.12 Connecting Analog Signal Sources

INFORMATION

Only C-413.2GA and .20A: Analog inputs are available on pins 1 and 3 of the **I/O** panel plug. Each analog input can be used for an external sensor or as the source for generating control values; refer to "Analog Input Signals" (p. 94) for details.

Tools and accessories

- Suitable signal source, see the pin assignment of the **I/O** panel plug (p. 295) for details
- Suitable cable, e.g., C-413.1IO I/O cable with open end (p. 296), available as an optional accessory (p. 11).

Connecting an analog signal source

➢ Connect an appropriate signal source to pin 1 or 3 of the **I/O** panel plug of the C-413.2GA or .20A.

5.13 Connecting a Device to the Analog Output

INFORMATION

Only C-413.2GA and .20A: Analog outputs are available on pins 5 and 7 of the **I/O** panel plug. Each analog output can be used to monitor the position, force or velocity of an axis or to control an external motor driver; refer to "Analog Output Signals" (p. 110) for details.

Tools and accessories

- Suitable measuring device or suitable motor driver,see the pin assignment of the **I/O** panel plug (p. 295) for details
- Suitable cable, e.g., C-413.1IO I/O cable with open end (p. 296), available as an optional accessory (p. 11).

Connecting a device to an analog output

➢ Connect an appropriate device to pin 5 or 7 of the **I/O** panel plug of the C-413.2GA or .20A.

6 Startup

6.1 General Notes on Startup

NOTICE

Electromagnetic disturbances

If a C-413 OEM board is operated without a housing, live parts are accessible. Electrical, magnetic and electromagnetic fields emitted by live parts can disturb the C-413 and/or the environment.

➢ Install a C-413 OEM board in a suitable housing before startup (p. 61).

➢ Make sure that the C-413 OEM board fulfills all requirements for electromagnetic compatibility after being installed in a housing.

NOTICE

Unexpected motion of the C-413

The C-413 can be configured with parameter settings so that the referencing move (p. 46) and/or the autozero procedure (p. 49) is run automatically after switching on or rebooting. If setup has not been prepared for the corresponding motion yet, the mechanics and/or the load attached to it could be damaged by collisions.

➢ Make sure that the connected mechanics can move over the entire travel range safely when the C-413 is switched on or rebooted.

➢ If you have configured the C-413 to do a referencing move and/or the autozero procedure automatically: Make sure that all system users have been informed about the configuration before the C-413 is switched on or rebooted.

NOTICE

Unexpected motion from lack of self-locking

Due to the lack of self-locking of the voice coil drive, mechanics connected to the C-413 can move unexpectedly in the following cases:

- Switching the C-413 off
- Rebooting the C-413 with the `RBT` command (p. 194) or with the corresponding functions of the PC software
- Switching servo mode off for the axis.

➢ Note: The C-413 switches servo mode off automatically when the axis is in overflow condition (p. 191) for more than 60 s.

Unexpected motion can result in damage to the mechanics and/or the load attached to it, e.g., from the moving part falling onto the hard stop.

➢ When the motion axis is aligned vertically: Run an autozero procedure for the axis (p. 49) so that the weight force of the moving mass is also compensated when servo mode is switched off.

- ➢ Before switching off or rebooting the C-413, take suitable precautionary measures to ensure that no unexpected motion is possible due to lack of self-locking of the voice coil drive. Examples of measures:
  - Moving to a "safe" position, e.g., the lower end of the travel range when the motion axis is aligned vertically
  - Installing a mechanical device to catch the moving part

NOTICE

Damage due to moving to the hard stop

Moving to the hard stop at a high velocity can result in damage to the mechanics.

The axis can move to the hard stop at a high velocity in the following cases:

- Motion is triggered in open-loop operation.
- Motion is triggered in closed-loop operation, and the control variable is the velocity or the force.

➢ Ensure that the hard stop is approached at low velocity.

NOTICE

Mechanics overheating

When a high control value remains set over a long period of time, the mechanics can heat up. Overheating can result in damage to the mechanics.

I^2t monitoring:

➢ Activate I2t monitoring (p. 51) to prevent the mechanics from overheating.

Closed-loop operation, the control variable is the position or the force:

To protect the mechanics, the C-413 switches servo mode off automatically for the axis, when the axis is in overflow state for more than 60 s (query with OVF? (p. 191)). Switching servo mode off reduces the absolute measure of the control value from the maximum to the value of the ***AutoZero Result*** parameter (ID 0x07000A03, refer to "Autozero Procedure for Compensating the Weight Force" (p. 49)).

Closed-loop operation, the control variable is the velocity:

If the axis is moved to the hard stop or blocked by an obstacle **and** the target velocity is zero (e.g., after the axis has stopped), the overflow state does **not** occur, and servo mode is **not** switched off automatically. Reduce the absolute measure of the control value as follows:

➢ Switch off servo mode for the axis manually.

or

1. Command slow motion of the axis away from the hard stop or obstacle.
2. Stop the axis while it is moving freely.

NOTICE

Mechanics oscillating

The optimal values of the servo control parameters of the C-413 depend on the selected control mode and the application. In particular the parameter values for direct control of a control variable strongly differ from the values that are required for control with a cascade structure. Unsuitable setting of the C-413's servo control parameters can cause the mechanics to oscillate. Oscillation can damage the mechanics and/or the load fixed to it.

- If the mechanics are oscillating (unusual operating noise), switch servo mode or the C-413 off immediately.
- Switch servo mode back on only after you have modified the servo control parameter settings.
- Check the values of the servo control parameters each time the control mode is changed.
- If you have configured the C-413 to switch on servo mode automatically when the C-413 is switched on or rebooted: Make sure that all system users have been informed about the configuration.

The ***Available Closed-Loop Control Modes*** parameter (ID 0x07030101) is intended to prevent inadvertent selection of a control mode where the servo control parameters of the C-413 are not adapted; refer to "Control Modes and Control Variables" (p. 28).

- Change the value of the ***Available Closed-Loop Control Modes*** parameter only if necessary.

INFORMATION

When the axis is in overflow state for more than 60 s (get with OVF? (p. 191)), the C-413 switches off the servo mode for the axis. Possible causes for the occurrence of the overflow state:

- The axis has not yet been referenced (query with FRF?).
- Axis oscillates
- When the control variable is the position or the velocity: The axis is blocked by an obstacle.
- When the control variable is the velocity or the force: The axis has reached the hard stop.

INFORMATION

The C-413 is switched off by disconnecting the power supply. Options:

- Pull the power cord out of the power socket.
- Pull the power cord out of the power adapter.
- Pull the power adapter connector out of the C-413.

6.2 Switching the C-413 On

INFORMATION

The C-413 does the following when switching on or rebooting:

- Initializing the sensor electronics in the mechanics
- Reading the ID chips of the sensors; refer to "Detecting the ID Chip" (p. 53) for details
- When you have connected mechanics to the switched on C-413: Switch the C-413 off and on again, or reboot the C-413 with the RBT command (p. 223) or with the corresponding function of the PC software.

Requirements

- ✓ You have read and understood the General Notes on Startup (p. 69).
- ✓ The C-413 has been installed properly (p. 59).

Switching the C-413 on

- ➢ Connect the power cord of the power supply with the power socket.

6.3 Establishing Communication

The procedure for PIMikroMove is described in the following.

INFORMATION

In the C-413, a USB UART module is used for the USB interface. This results in the following:

- A baud rate setting is necessary for the USB interface.
- If the controller is connected via the USB connection and switched on, the USB interface is also shown as a virtual COM port in the PC software.

To successfully establish communication via USB, the baud rates of the PC and C-413 must be identical. The PIMikroMove PC software, which PI recommends for initial startup of the C-413, therefore automatically adapts the baud rate of the PC to the current baud rate of the C-413.

Only when the PC software offers the selection of the baud rate of the PC when communication is established via USB:

- ➢ Adapt the baud rate of the PC to the current baud rate of the C-413.

Requirements

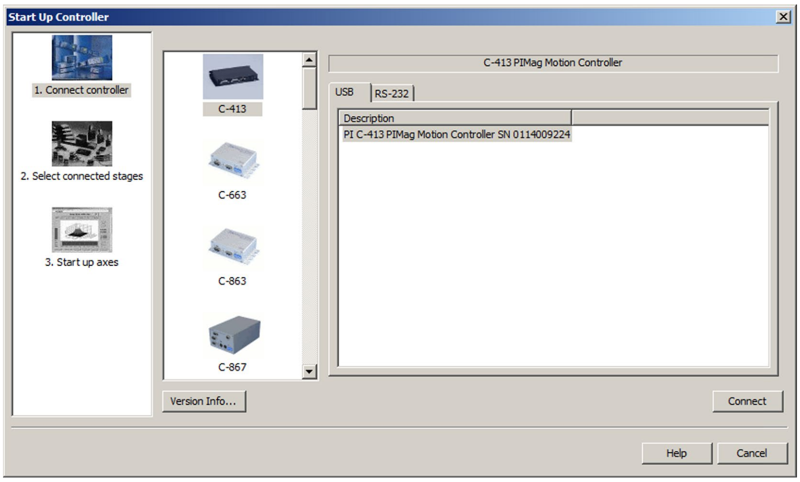
- ✓ You have read and understood the General Notes on Startup (p. 69).
- ✓ The C-413 is connected to the USB interface of the PC (p. 65).
- ✓ The C-413 is switched on (p. 71).
- ✓ The PC is switched on.
- ✓ The required software and USB drivers are installed on the PC (p. 59).
- ✓ You have read and understood the manual for the PC software used. The links to the software manuals are in the A000T0081 file on the PI software data storage medium.

Establishing communication

1. Start PIMikroMove.

   The ***Start up controller*** window opens with the ***Connect controller*** step.

– If the ***Start up controller*** window does not open automatically, select the ***Connections > New...*** menu item in the main window.



2. Select ***C-413*** in the controller selection field.
3. Select the ***USB*** tab on the right-hand side of the window.
4. Select the connected C-413 in the ***USB*** tab.
5. Select ***Connect*** to establish communication.

If communication was established successfully, PIMikroMove guides you through configuring the C-413 for the connected mechanics; refer to "Starting Motion" (p. 73).

➢ If communication could not be established, look for a solution to the problem in "Troubleshooting" (p. 281).

6.4 Starting Motion

PIMikroMove is used to move the mechanics in the following. The program guides you through the following steps so that you do not have to deal with the respective GCS commands:

- Doing a referencing move: Refer to "Referencing" (p. 46) for details.
- Running an autozero procedure: Refer to "Autozero Procedure for Compensating the Weight Force" (p. 49) for details.

NOTICE

Unexpected motion from lack of self-locking
Due to the lack of self-locking of the voice coil drive, mechanics connected to the C-413 can move unexpectedly in the following cases:

- Switching the C-413 off
- Rebooting the C-413 with the `RBT` command (p. 194) or with the corresponding functions of the PC software
- Switching servo mode off for the axis.

- ➢ Note: The C-413 switches servo mode off automatically when the axis is in overflow condition (p. 191) for more than 60 s.

Unexpected motion can result in damage to the mechanics and/or the load attached to it, e.g., from the moving part falling onto the hard stop.

- ➢ When the motion axis is aligned vertically: Run an autozero procedure for the axis (p. 49) so that the weight force of the moving mass is also compensated when servo mode is switched off.
- ➢ Before switching off or rebooting the C-413, take suitable precautionary measures to ensure that no unexpected motion is possible due to lack of self-locking of the voice coil drive. Examples of measures:
  - Moving to a "safe" position, e.g., the lower end of the travel range when the motion axis is aligned vertically
  - Installing a mechanical device to catch the moving part

INFORMATION

When the axis is in overflow state for more than 60 s (get with OVF? (p. 191)), the C-413 switches off the servo mode for the axis. Possible causes for the occurrence of the overflow state:

- The axis has not yet been referenced (query with FRF?).
- The axis oscillates.
- The axis is blocked by an obstacle.
- When the control variable is the velocity: The axis has reached the hard stop.

INFORMATION

In the following, work is done with the default settings of the C-413:

- Control variable: Position (PID_Pos_Vel = PID position control with velocity control, ID 7)
- Further selectable control variables:
  - Velocity (PID_Vel = Direct PID velocity control, ID 6)
  - Position (PID_Pos = direct PID position control, ID 1)

To show a change in the control variable, PID_Pos_Vel is used at the beginning and then switched to PID_Vel in the following.

- ➢ Note that changing the control variable from position to velocity changes the behavior of the axis:
  - Position as control variable: The target position is approached and maintained. The motion is then finished.
  - Velocity as control variable: The axis moves at the target velocity up to the hard stop. If the target velocity at the hard stop is not set to zero, the drive warms up and the overflow state may occur.

Requirements

- ✓ You have read and understood the General Notes on Startup (p. 69).
- ✓ PIMikroMove is installed on the PC (p. 59).

- ✓ You have read and understood the PIMikroMove manual.
- ✓ You have installed the mechanics in the same way as they will be used in your application (corresponding load, orientation, and fixing).
- ✓ You have connected the C-413 to the mechanics (p. 64).
- ✓ PIMikroMove has established communication between the C-413 and the PC (p. 72).

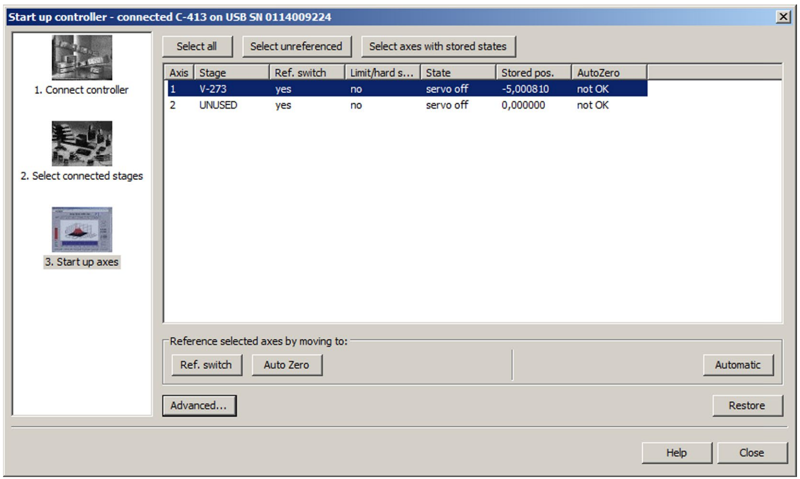
Starting motion with PIMikroMove

The figures in the following instructions show an example in which axis 2 is **not** used.

When communication has been successfully established between the C-413 and the PC (p. 72), the ***Start up controller*** window automatically goes to the ***Start up axes*** step.

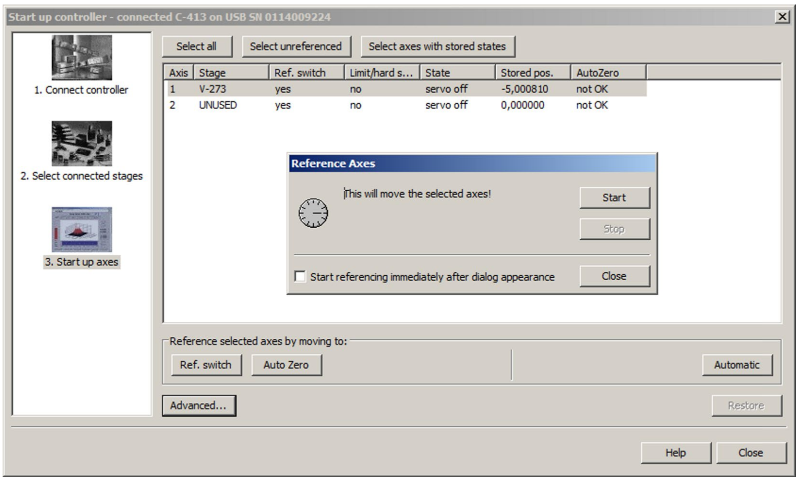
1. In the ***Start up axes*** step, execute the referencing move so that the controller knows the absolute axis position. Proceed as follows for each axis that is connected:

   a) Mark the axis in the list.

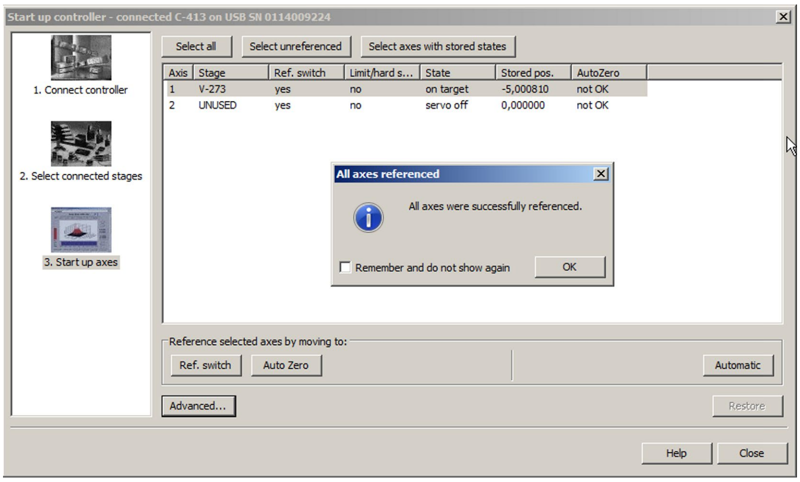


   b) Select ***Ref. switch*** or ***Automatic***. The ***Reference Axes*** dialog opens.

   c) In the ***Reference Axes*** dialog, start the referencing move with ***Start***.



d) After a successful referencing move, select ***OK***.

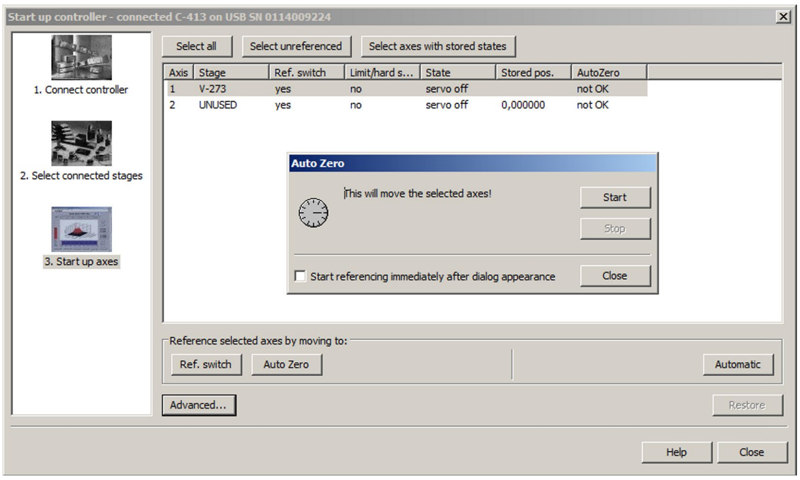


2. In the ***Start up axes*** step, run the autozero procedure. Proceed as follows for each axis that is connected:

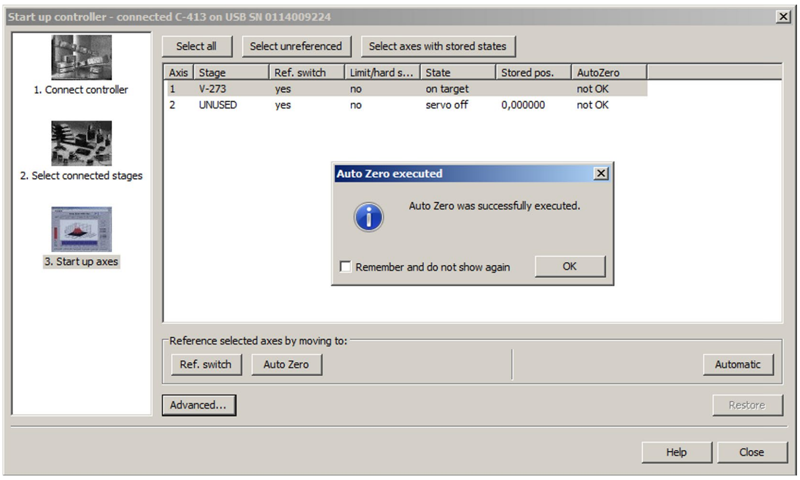
   a) Mark the axis in the list.

   b) Select ***Auto Zero***. The ***Auto Zero*** dialog opens.

   c) In the ***Auto Zero*** dialog, start the autozero procedure with ***Start***.



d) After a successful autozero procedure, select ***OK***.

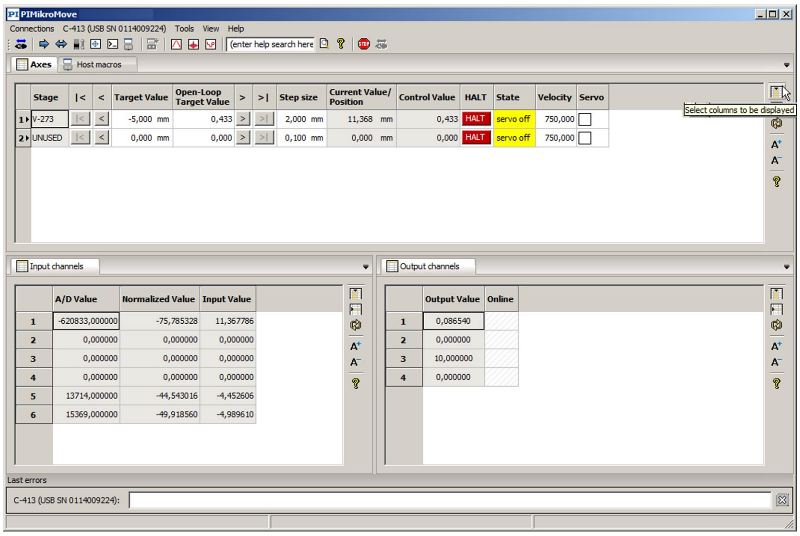


3. In the ***Start up controller*** window, select ***Close***.

   The main window of PIMikroMove opens.

4. In the main window of PIMikroMove, go to the ***Axes*** card and display the ***Closed-Loop Control Mode*** column, which shows the selected control mode:

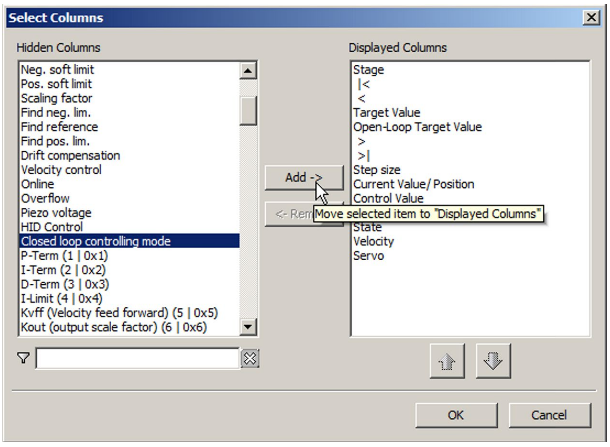
   a) On the right margin of the ***Axes*** card, select (***Select columns to be displayed***).



The ***Select Columns*** dialog opens.

   b) In the ***Select Columns*** dialog, go to the ***Hidden Columns*** area and select the ***Closed-Loop Control Mode*** line.

c) In the ***Select Columns*** dialog, select ***Add ->***.

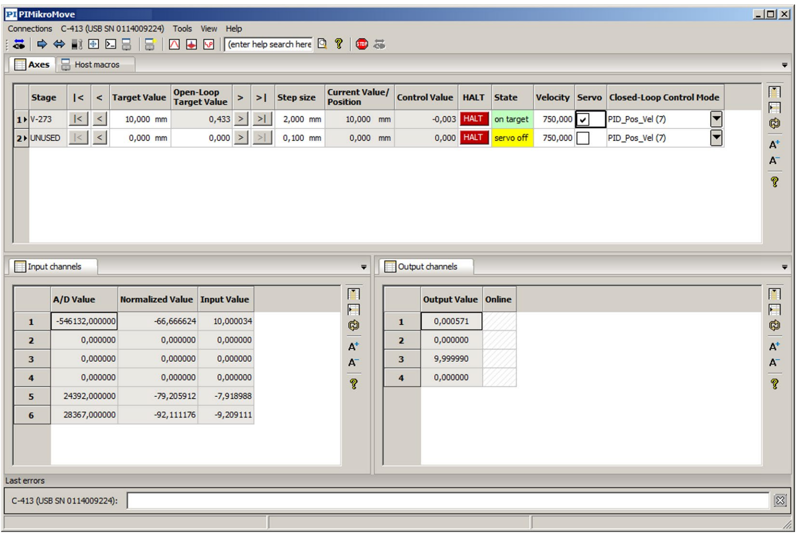


d) Close the ***Select Columns*** dialog with ***OK***.

5. On the ***Axes*** card in the main window of PIMikroMove, switch on the servo mode. Proceed as follows for each axis that is to be moved:

a) Read the selected control mode in the ***Closed-Loop Control Mode*** column.

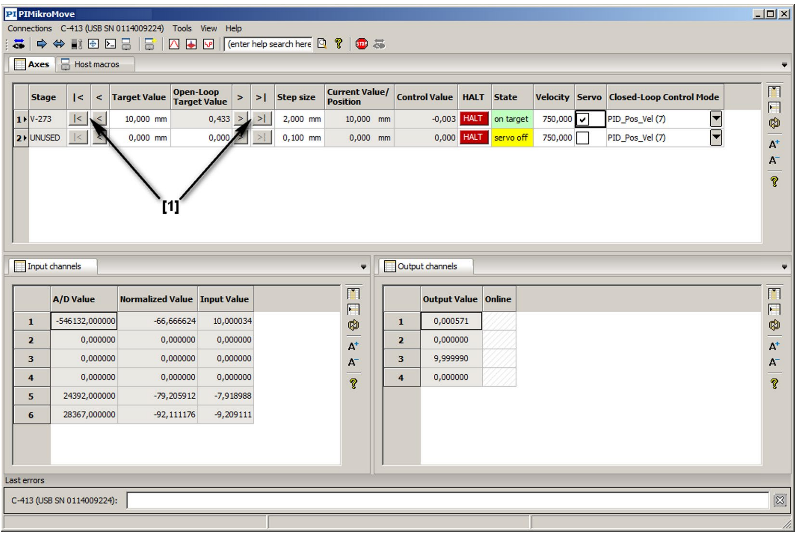
b) Switch on the servo mode by marking the check box in the ***Servo*** column.



In the example shown in the figure above, axis 1 is in closed-loop operation, and the control variable is the position (the PID_Pos_Vel (7) control mode is selected by default).

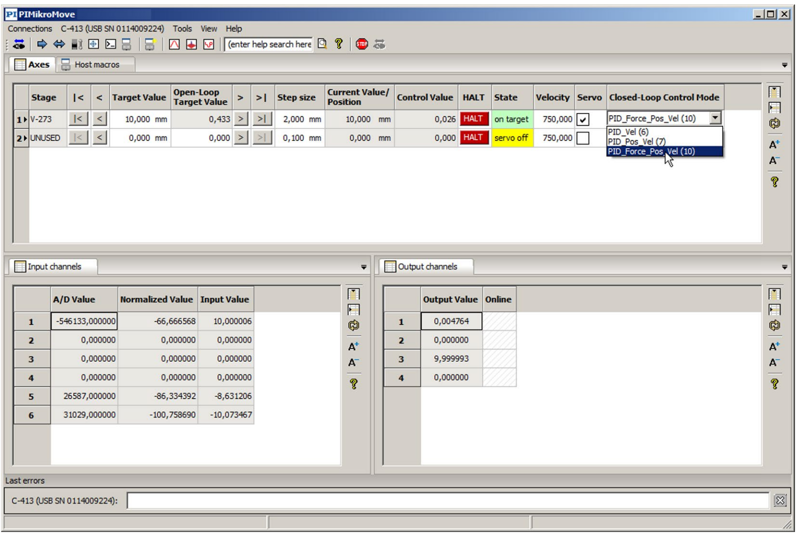
6. Start a few test motions to position the axis.

   On the ***Axes*** card in the main window of PIMikroMove, you can execute, for example, motions of a particular distance (specification in ***Step size*** column) or to the limits of the travel range with the corresponding arrow keys [1] for the axis.



7. If you want to change the control variable for the axis:

   – In the main window of PIMikroMove, select the new control mode in the ***Closed-Loop Control Mode*** column of the ***Axes*** card.

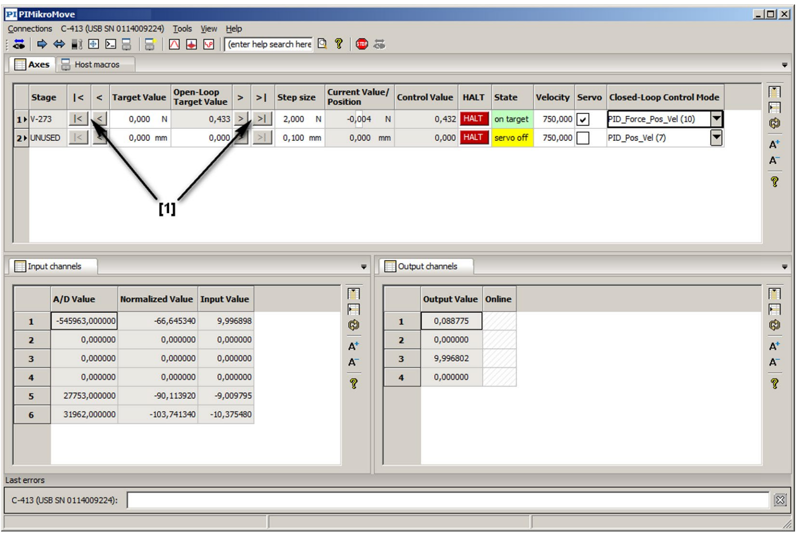


   Default setting: The PID_Vel (6), PID_Pos_Vel (7) and PID_Force_Pos_Vel (10) control modes can be selected. When switching between these control modes, it is not necessary to adapt the servo control parameters of the C-413.

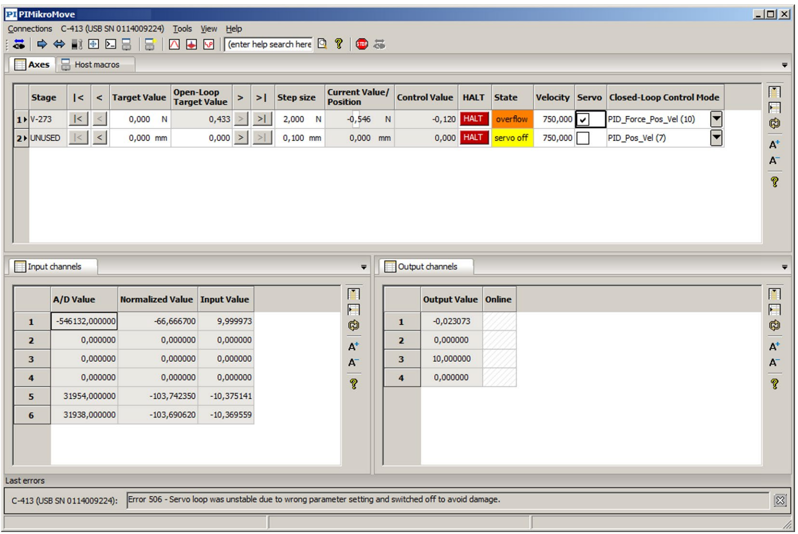
   In the example shown in the figure above, PID_Force_Pos_Vel (10) is selected. The new control variable is therefore the force. The unit sign on the ***Axes*** card is adjusted automatically.

8. Start a few test motions in which the axis is to apply different forces.

   On the ***Axes*** card in the main window of PIMikroMove, you can change, for example, the applied force by a particular amount (specification in the ***Step size*** column) or have the maximum force applied with the corresponding arrow keys [1] for the axis.

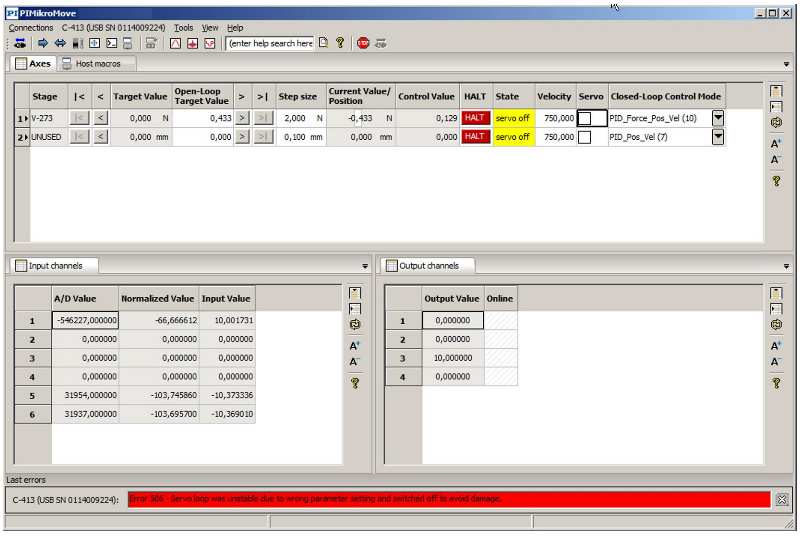
   When the overflow state (p. 191) occurs for the axis (display in the ***State*** column):

   a) When the axis is at the hard stop: Change the value for the target force so that the axis moves away from the hard stop.

   b) Have a counterforce applied to the axis in the amount of the target force.

When the C-413 has switched off the servo mode for the axis because the axis was in the overflow state for more than 60 s, an error code is set:



c) Switch the servo mode for the axis back on by marking the check box in the line of the axis in the ***Servo*** column.

d) Prevent the overflow state from occurring again, see steps a) and b).

7 Operation

7.1 Data Recorder

7.1.1 Data Recorder Properties

The C-413 contains a real-time data recorder. The data recorder can record several input and output signals (e.g., current position, sensor input, output current) from different data sources (e.g., logical axis, input and output signal channel).

The recorded data is stored temporarily in up to 8 data recorder tables. Each data recorder table contains the data of one data source. The total number of points in the data recorder tables is §§§ mm. These points are distributed evenly among the data recorder tables used and therefore determine the size of the table(s). If you use the maximum number of 8 data recorder tables for example, each data recorder table will consist of 1 points.

You can configure the data recorder e.g., by defining the record options, the data sources, and the number of data recorder tables.

7.1.2 Configuring the Data Recorder

Reading out general information on the data recorder

- Send the `HDR?` command (p. 181).

  The available record options and trigger options as well as information on additional parameters and commands for data recording are displayed.

Setting the number of data recorder tables

- Send the `TNR?` command (p. 210) to read the number of available data recorder tables.

  The response shows the value of the ***Data Recorder Channel Number*** parameter (ID 0x16000300).

- Set the number of data recorder tables by assigning the ***Data Recorder Chan Number*** parameter (ID 0x16000300) a value between *1* and *8* with the `SPA` command (volatile memory) (p. 199) or the `SEP` command (nonvolatile memory) (p. 197).

The size of the data recorder tables is set along with the number (p. 83). The total number of points available is specified by the ***Max Points*** parameter (ID 0x16000200).

Configuring recording

You can assign the data sources and record options to the data recorder tables.

- Send the `DRC?` command (p. 174) to read out the current configuration. Data recorder tables with the record option 0 are deactivated, i.e., nothing is recorded. In the default setting, the C-413 records the current position of axis 1 in data recorder table 1.
- Configure the data recorder with the `DRC` command (p. 172).

You can specify how the recording is to be triggered.

- Query the current trigger option with `DRT?` (p. 177).
- Change the trigger option with the `DRT` command (p. 176). The trigger option applies to all data recorder tables with a record option not set to 0.

  When you have selected the "External trigger" trigger option with `DRT`, configure and activate the trigger input for the specified digital input line with the `CTI` (p. 163) and `TRI` (p. 211) commands. Refer to "Data Recorder" Trigger Mode - Starting Data Recording" (p. 92) for further information.

INFORMATION

The configuration settings with the `DRC`, `DRT`, `CTI` and `TRI` commands are only written to the volatile memory and are lost when the C-413 is switched off or rebooted.

Setting sampling interval

- Send the `RTR?` command to read the record table rate of the data recorder.

  The response shows the value of the ***Data Recorder Table Rate*** parameter (ID 0x16000000). The parameter indicates after how many servo cycles each data point is recorded. The default value is one servo cycle.
- Set the record table rate by changing the ***Data Recorder Table Rate*** parameter (ID 0x16000000):
  - Write the new record table rate to the volatile memory with the `RTR` command (p. 195).

  \- or -
  - Change the record table rate in the volatile memory with the `SPA` command (p. 199) or in the nonvolatile memory with the `SEP` command (p. 197).

  As the record table rate increases, you increase the maximum duration of the data recording.

INFORMATION

You can also read the parameters of the data recorder in the volatile memory with the `SPA?` command and in the nonvolatile memory with the `SEP?` command.
Except for the configuration settings for recording, you can write parameter values from the volatile memory to the nonvolatile memory with the `WPA` command (p. 229). This makes them default values that remain valid even after the C-413 is rebooted.

7.1.3 Starting the Recording

- Start the recording with the trigger option set with `DRT`.

Regardless of the trigger option set, the data recording is always triggered in the following cases:

- Start of a step response measurement with `STE` (p. 202)
- Start of an impulse response measurement with `IMP` (p. 187)
- Start of the wave generator with `WGO` (p. 224), bit 0

- When of the wave generator is running: Start of data recording with `WGR` (p. 227)

Data recording is always done for all data recorder tables with a record option not set to 0. It ends when the data recorder tables are full.

7.1.4 Reading Recorded Data

INFORMATION

Reading the recorded data can take some time, depending on the number of data points.
The data can also be read while data is being recorded.

- Read out the last recorded data with the `DRR?` command (p. 175).

  The data is output in the GCS array format (refer to the SM146E user manual).
- Query the number of points in the last recording with the `DRL?` command (p. 174).

7.2 Digital Output Signals

The digital outputs of the C-413 are available on the **I/O** connector.

- Query the number of digital output lines available on the C-413 with the `TIO?` command (p. 209).

External devices can be triggered via the digital outputs of the C-413. Details and examples for coupling the trigger output to the axis motion can be found in this section.

7.2.1 Commands for Digital Outputs

The following commands are available for the use of digital outputs:

Command	Syntax	Function
`CTO`	CTO {<TrigOutID> <CTOPam> <Value>}	Configures the conditions for the trigger output. Couples the trigger output to the axis motion.
`CTO?`	CTO? [{<TrigOutID> <CTOPam>}]	Queries the current configuration of the trigger output.
`DIO`	DIO {<DIOID> <OutputOn>}	Switches digital output lines directly to the low or high state, either separately or all lines at once. Should not be used for output lines where the trigger output is activated with `TRO`.
`TRO`	TRO {<TrigOutID> <TrigMode>}	Activates or deactivates the trigger output conditions set with `CTO`. Default: Trigger output deactivated.
`TRO?`	TRO? [{<TrigOutID>}]	Queries the current activation state of the trigger output conditions set with `CTO`.

The settings for triggering a device via the digital outputs can be transferred to the C-413 with the following command (CTO + max. 12 arguments):

`CTO {<TrigOutID> <CTOPam> <Value>}`

- `<TrigOutID>` is one digital output line of the controller.
- `<CTOPam>` is the CTO parameter ID in decimal format.
- `<Value>` is the value to which the CTO parameter is set.

The following trigger modes (<Value>) can be set for <CTOPam> = 3:

<Value>	Trigger mode	Short description
0	Position Distance	Once the axis has moved a specified distance, a trigger pulse is output. Optionally, start and stop values can be defined to limit triggering to one position range and one particular direction of motion (negative or positive).
2 (default)	On Target	The on-target state of the axis selected is output at the selected trigger output (p. 88).
3	MinMax Threshold	The selected digital output line is active when the position of the selected axis is within a specified band (p. 89).
6	In Motion	The selected digital output line is active as long as the selected axis is in motion.

In addition, the polarity (active high / active low) of the signal at the digital output can be set (p. 90).

INFORMATION

The settings for the configuration of the digital output lines can only be modified in the volatile memory of the C-413. After the C-413 has been switched on or rebooted, factory default settings are activated.

7.2.2 Configuring the "Position Distance" Trigger Mode

The *Position Distance* trigger mode is suitable for scanning applications. A trigger pulse is output as soon as the axis has moved along the distance set with CTO parameter ID = 1 (TriggerStep). The pulse width is one servo cycle (p. 139).

The unit of the distance (TriggerStep) depends on the setting of the C-413; refer to "Control Modes and Control Variables" (p. 28).

1. Configure the digital output line (<TrigOutID>) to be used as the trigger output:
   – Send `CTO <TrigOutID> 2 A`, where *A* indicates the axis to be moved.
   – Send `CTO <TrigOutID> 3 0`, where 0 determines the *Position Distance* trigger mode.
   – Send `CTO <TrigOutID> 1 S`, where *S* indicates the distance.

2. If you want to activate the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example:

A pulse on digital output line 1 is output each time axis 1 of the mechanics covers a distance of 0.1 mm. The unit of the position is millimeters.

➢ Send:

`CTO 1 2 1`

`CTO 1 3 0`

`CTO 1 1 0.1`

`TRO 1 1`

"Position Distance" trigger mode with start and stop values for positive motion direction of the axis

Optionally, you can define start and stop values for limiting the range and for specifying the motion direction of the axis (positive or negative).

INFORMATION

If start and stop values have the same value, they are ignored.

If the direction of motion is reversed before the axis position has reached the stop value, trigger pulses continue to be output.

1. Configure the digital output line (<TrigOutID>) to be used as the trigger output:
   - Send `CTO <TrigOutID> 2 A`, where *A* indicates the axis to be moved.
   - Send `CTO <TrigOutID> 3 0`, where 0 determines the *Position Distance* trigger mode.
   - Send `CTO <TrigOutID> 1 S`, where *S* indicates the distance.
   - Send `CTO <TrigOutID> 8 Start`, where *Start* indicates the start value.
   - Send `CTO <TrigOutID> 9 Stop`, where *Stop* indicates the stop value.
2. If you want to activate the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example

A pulse on digital output line 1 is output each time axis 1 of the mechanics covers a distance of 0.1 mm, as long as axis 1 is moving in a positive direction within the range of 0.2 mm to 0.55 mm (start value < stop value). The unit of the position is millimeters.

➢ Send:

`CTO 1 2 1`

`CTO 1 3 0`

`CTO 1 1 0.1`

`CTO 1 8 0.2`

`CTO 1 9 0.55`

`TRO 1 1`

"Position Distance" trigger mode with start and stop values for negative motion direction of the axis

The following shows the above example with swapped start and stop values. Triggering is done when the axis is moving in a negative direction (stop value < start value) in the range between 0.55 mm and 0.2 mm.

Example:

➢ Send:

`CTO 1 2 1`

`CTO 1 3 0`

`CTO 1 1 0.1`

`CTO 1 8 0.55`

`CTO 1 9 0.2`

`TRO 1 1`

7.2.3 Configuring the "On Target" Trigger Mode

The on-target state of the axis selected (p. 45) is output at the selected trigger output in *On Target* trigger mode.

1. Configure the digital output line (<TrigOutID>) to be used as the trigger output:
   - Send `CTO <TrigOutID> 2 A`, where *A* indicates the axis to be moved.
   - Send `CTO <TrigOutID> 3 2`, where *2* specifies the *On Target* trigger mode.
2. If you want to activate the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example:

The on-target state of axis 1 is to be output on the digital output line 1.

➢ Send:

`CTO 1 2 1`

`CTO 1 3 2`

`TRO 1 1`

7.2.4 Configuring the "MinMax Threshold" Trigger Mode

The selected digital output line is active in *MinMax Threshold* trigger mode when the position of the selected axis is within a specified band (p. 89).

The unit of the position values for limiting the band depends on the settings of the C-413; refer to "Control Modes and Control Variables" (p. 28).

1. Configure the digital output line (<TrigOutID>) to be used as the trigger output:
   - Send `CTO <TrigOutID> 2 A`, where *A* indicates the axis to be moved.
   - Send `CTO <TrigOutID> 3 3`, where 3 determines the *MinMax Threshold* trigger mode.
   - Send `CTO <TrigOutID> 5 Min`, where *Min* indicates the position value for the lower limit of the band.
   - Send `CTO <TrigOutID> 6 Max`, where *Max* indicates the position value for the upper limit of the band.
2. If you want to activate the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example:

Digital output line 1 is to be active if the current position of axis 1 of the mechanics is in the range between 0.2 mm and 0.55 mm. The unit of the position is millimeters.

- Send:

  `CTO 1 2 1`

  `CTO 1 3 3`

  `CTO 1 5 0.2`

  `CTO 1 6 0.55`

  `TRO 1 1`

7.2.5 Configuring the "In Motion" Trigger Mode

The motion state of the selected axis is output at the selected trigger output in the *In Motion* trigger mode. The line is active as long as the selected axis is in motion.

The motion state can also be read with the `#5` (p. 149) and `SRG?` (p. 201) commands.

INFORMATION

If the axis is in motion, then bit 13 of the status register is set.

1. Configure the digital output line (<TrigOutID>) to be used as the trigger output:
   - Send `CTO <TrigOutID> 2 A`, where *A* indicates the axis to be moved.
   - Send `CTO <TrigOutID> 3 6`, where 6 specifies the *In Motion* trigger mode.

2. If you want to activate the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example:

Digital output line 1 is to be active if axis 1 of the mechanics is in motion.

➢ Send:

`CTO 1 2 1`

`CTO 1 3 6`

`TRO 1 1`

7.2.6 Setting Signal Polarity

The polarity of the signal at the digital output which is used for triggering can be selected with the *Polarity* CTO parameter. The polarity can have the following values:

- active high = 1 (default setting)
- active low = 0

➢ Configure the digital output line (<TrigOutID>) to be used as the trigger output:

– Send `CTO <TrigOutID> 7 P`, where *P* indicates the polarity.

Example:

The signal polarity for digital output line 1 is to be set to active low.

➢ Send:

`CTO 1 7 0`

7.3 Digital Input Signals

The digital inputs of the C-413 are available on the **I/O** connector.

➢ Query the number of digital input lines available on the C-413 with the `TIO?` command (p. 209).

The data recorder and the wave generator can be triggered via the digital inputs of the C-413. Details and examples can be found in this section.

7.3.1 Commands for Digital Inputs

The following commands are available for the use of digital inputs:

Command	Syntax	Function
`CTI`	CTI {<TrigInID> <CTIPam> <Value>}	Configures the trigger input.
`CTI?`	CTI? [{<TrigInID> <CTIPam>}]	Queries the current configuration of the trigger input.

Command	Syntax	Function
`DIO?`	DIO? [{<DIOID>}]	Queries the state of the digital input lines (low or high).
`TRI`	TRI {<TrigInID> <TrigInMode>}	Activates or deactivates the trigger input configuration made with `CTI`. Default: Configuration deactivated.
`TRI?`	TRI? [{<TrigInID>}]	Queries the current activation state of the configuration made with `CTI`.

The settings for trigger input via the digital inputs can be transferred to the C-413 with the following command (CTI + max. 12 arguments):

`CTI {<TrigInID> <CTIPam> <Value>}`

- `<TrigInID>` is one digital input line of the controller.
- `<CTIPam>` is the CTI parameter ID in decimal format.
- `<Value>` is the value to which the CTI parameter is set.

The following trigger modes (<Value>) can be set for <CTIPam> = 3:

<Value>	Trigger mode	Short description
0 (default)	No triggering	-
2	Data Recorder	The digital input line triggers a recording by the data recorder. Further condition: With DRT (p. 176), the "External trigger" trigger option must be set and the input line selected with DRT must match the input line selected with CTI.
4	Wave Generator	The digital input line starts and interrupts the wave generator output. The specified trigger type determines the output behavior of the wave generator. Further condition: For the selected wave generators, the start mode "Start via external trigger signal" (bit 1) must be set with WGO (p. 224).

In addition, the polarity (active high / active low) of the signal at the digital input can be set.

INFORMATION

The settings for the configuration of the digital input lines can only be modified in the volatile memory of the C-413. After the C-413 has been switched on or rebooted, factory default settings are activated.

7.3.2 "Data Recorder" Trigger Mode - Starting Data Recording

In the *Data Recorder* trigger mode, the selected digital input line triggers a recording by the data recorder. The setting for the trigger type determines how the triggering takes place. Possible trigger types (CTIPam 1):

- 0 = Edge triggered (default); triggering upon state transition of the digital input line. The activating state transition can be low --> high or high --> low (depends on the signal polarity set (CTIPam 7)).
- 1 = Level triggered; triggering when the digital input line is in the active state (high or low; depends on the signal polarity set (CTIPam 7)).

In addition to the settings made with CTI and TRI, the "External trigger" trigger option must be set with the DRT command (p. 176). The input line used for DRT must match the input line configured with CTI.

Starting data recording in "Data Recorder" trigger mode

1. Use CTI to configure the <TrigInID> digital input line that is to be used as the trigger input:
   - Send `CTI <TrigInID> 3 2`, where 2 determines the *Data Recorder* trigger mode.
   - Send `CTI <TrigInID> 1 T`, whereby *T* determines the trigger type (0 or 1).
   - Send `CTI <TrigInID> 7 P`, where *P* determines the signal polarity (0 = active low, 1 = active high (default)).
2. Activate the trigger configuration of the <TrigInID> digital input line:
   - Send `TRI <TrigInID> 1`.
3. Configure the data recorder for starting the recording with the <TrigInID> digital input line:
   - Send `DRT 0 3 <TrigInID>`, where *0* specifies the data recorder table that the recording is to be started for (0 = all tables), and *3* determines the "External trigger" trigger option.
   - Optional: Set the data sources and record options with the DRC command (p. 172).

   Refer to "Data Recorder" (p. 83) for detailed information.
4. Start the data recording:
   - Activate the <TrigInID> digital input line according to the settings for trigger type and signal polarity.

   A recording cannot be triggered again until the recording in progress has ended (i.e. when the data recorder tables are full); this also requires the "External trigger" option to be set again with DRT (see step 3).

Example:

The data recording is to be started when the signal on digital input line 1 changes from the "low" state to the "high" state.

➢ Send:

```
CTI 1 3 2
CTI 1 1 0
CTI 1 7 1
TRI 1 1
DRT 0 3 1
```

7.3.3 "Wave Generator" Trigger Mode – Starting the Wave Generator Output

The selected digital input line starts/interrupts the output of the selected wave generator (CTIPam 13) in *Wave Generator* trigger mode.

In addition to the settings made with CTI and TRI, the start mode "Start via external trigger signal" (bit 1) must be set for the selected wave generator with the WGO command (p. 224).

The output behavior of the wave generator depends on the trigger type setting made with CTI. Possible trigger types (CTIPam 1):

- 0 = Edge triggered (default): Each activating state transition of the digital input line triggers output of one point in the wave table. The corresponding number of activating state transitions is required to output a point when an output rate of > 1 is set with the WTR command (p. 231). The activating state transition can be low --> high or high --> low (depends on the signal polarity set (CTIPam 7)).
- 1 = Level triggered: The wave generator outputs the points in the wave table when the digital input line is in the active state. Wave generator output is interrupted when the digital input line is not in the active state. The active state can be high or low (depends on the signal polarity set (CTIPam 7)).

Starting the wave generator output in the "Wave Generator" trigger mode

1. Use CTI to configure the <TrigInID> digital input line to be used as trigger input:
   - Send `CTI <TrigInID> 3 4`, where 4 determines the *Wave Generator* trigger mode.
   - Send `CTI <TrigInID> 1 T`, where *T* determines the trigger type (0 or 1).
   - Send `CTI <TrigInID> 13 W`, where *W* determines the wave generator (bit-coded specification of several wave generators possible).
   - Send `CTI <TrigInID> 7 P`, where *P* determines of the signal polarity (0 = active low, 1 = active high (default)).
2. Activate trigger configuration of the <TrigInID> digital input line:
   - Send `TRI <TrigInID> 1`.
3. Configure each wave generator selected with CTI as follows:
   a) Create the waveform in a wave table with the WAV command (p. 217).
   b) Connect the wave generator to the wave table created with the WSL command (p. 230).

c) Optional: Limit the number of waveform output cycles with the WGC command (p. 223).

Refer to "Wave Generator" (p. 120) for detailed information.

4. Start/interrupt output of the wave generators selected with CTI as follows:
   - Send `WGO F 0x2`, where *F* specifies the wave generator and *0x2* determines the start mode "Start via external trigger signal" (bit 1; start mode specified here in hexadecimal format).
   - Activate/deactivate the <TrigInID> digital input line according to the trigger type and signal polarity settings.

7.4 Analog Input Signals

This section describes how to use the analog inputs that are present on the **I/O** panel plug (p. 295) of the C-413.2GA and C-413.20A models (input signal channels 5 and 6).

7.4.1 Utilization Types for Analog Inputs

The following components can be connected to an analog input of the C-413:

- External position or force sensor
- Control source for the axis

The following figure shows the block diagram for the two types of use.

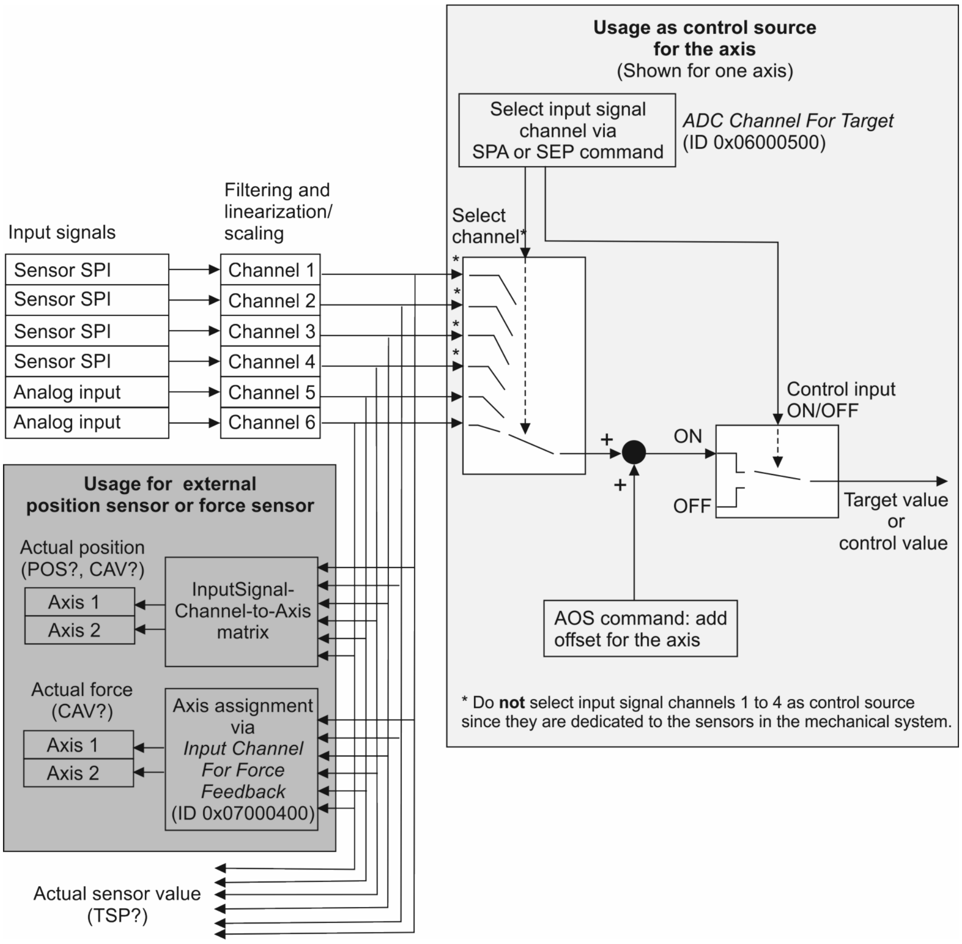


Figure 20: Block diagram for using an analog input

The analog inputs on the **I/O** panel plug can be addressed as input signal channels 5 and 6 in the firmware of the C-413 (p. 13).

INFORMATION

- Use the complete voltage range of an analog input (–10 to +10 V) to achieve maximum resolution.

The following steps are necessary to be able to use an analog input:

1. Scale the analog input to suitable position or force values (p. 98) independently of how it is used.
2. Configure the C-413 for the selected type of use of the analog input:
   - External sensor (p. 103)
   - Control source for the axis (p. 107)

An analog input that is not used must be deactivated (p. 109).

7.4.2 Commands and Parameters for Analog Inputs

Commands

The following commands are available for using an analog input:

Command	Syntax	Function
`AOS`	AOS {<AxisID> <Offset>}	Writes an axis-related offset for using the analog input as a control source to the volatile memory.
`AOS?`	AOS? [{<AxisID>}]	Gets the current offset value from the volatile memory.
`TAD?`	TAD? [{<InputSignalID>}]	Gets the current value of the analog/digital converter at the analog input (dimensionless).
`TNS?`	TNS? [{<InputSignalID>}]	Gets the value at the analog input after the electronics linearization (normalized value, dimensionless).
`TSC?`	TSC?	Gets the total number of input signal channels (value of the ***Number Of Input Channels*** parameter)
`TSP?`	TSP? [{<InputSignalID>}]	Gets the value at the analog input after the mechanics linearization (scaled value, in physical units).

Parameters

The following parameters are available for configuring an analog input:

Parameter	Description and Possible Values
Sensor Mech. Correction 1 ID 0x02000200	Offset (0 order coefficient) of the polynomial for mechanics linearization (p. 21) Required to scale the analog input to suitable position or force values; refer to "Scaling an Analog Input" (p. 98) for further information.
Sensor Mech. Correction 2 ID 0x02000300	Gain (1st order coefficient) of the polynomial for mechanics linearization (p. 21) Required to scale the analog input to suitable position or force values; refer to "Scaling an Analog Input" (p. 98) for further information.

Parameter	Description and Possible Values
Digital Filter Type ID 0x05000000	Settings for digital filtering after the A/D conversion; refer to "Processing Input Signal Channels" (p. 21) for further information
Digital Filter Bandwidth ID 0x05000001	
ADC Channel for Target ID 0x06000500	Input signal channel for control source Specifies the identifier of the input signal channel that is to be used as a control source for the axis. When the parameter has the value 0, no analog input is connected with the axis as a control source. Refer to "Using as a Control Source" (p. 107) for further information.
Analog Target Offset ID 0x06000501	Axis-related offset for analog input The offset is only effective when an input signal channel of the C-413 is connected to the axis as a control source via the ***ADC Channel for Target*** parameter (ID 0x06000500). The value of the ***Analog Target Offset*** parameter can also be set in the volatile memory with the `AOS` command. Refer to "Using as a Control Source" (p. 107) for further information.
Position Range Limit Min 0x07000000	Minimum and maximum commandable position in closed-loop operation, minimum and maximum commandable force in closed-loop and open-loop operation Refer to "Generating Control Values" (p. 24) for further information.
Position Range Limit Max 0x07000001	
Force Range Limit min 0x07000005	
Force Range Limit max 0x07000006	
Input Channel For Force Feedback ID 0x07000400	Axis allocation of force sensors Specifies the identifier of the input signal channel that is to be used as an input for the force sensor of the axis. When a force sensor is connected to an input signal channel, the coefficients of the input matrix (p. 16) must have the value zero for this channel. Refer to "Allocating Axes to Channels" (p. 16).for further information.
Position from Sensor 1 ID 0x07000500	Axis allocation of position sensors The parameters specify the coefficients of the input matrix (p. 16) that is intended for the allocation of position sensors to axes. Refer to "Allocating Axes to Channels" (p. 16) for further information.
Position from Sensor 2 ID 0x07000501	

Parameter	Description and Possible Values
Position from Sensor 3 ID 0x07000502	
Position from Sensor 4 ID 0x07000503	
Position From Sensor 5 ID 0x07000504	
Position From Sensor 6 ID 0x07000505	
Number of Input Channels ID 0x0E000B00	Total number of input signal channels The number of available analog inputs is the difference between the total number of input signal channels and the number of sensor channels (***Number Of Sensor Channels***, ID 0x0E000B03).
Number of Sensor Channels ID 0x0E000B03	Number of sensor channels Input signal channels that are only intended for sensors; inputs via the **Motor & Sensor** sockets (p. 294).

7.4.3 Scaling an Analog Input

Before an analog input can be used with an external sensor or as a control source, the input signal must be assigned suitable position or force values through scaling.

The following block diagram shows the scaling of an analog input.

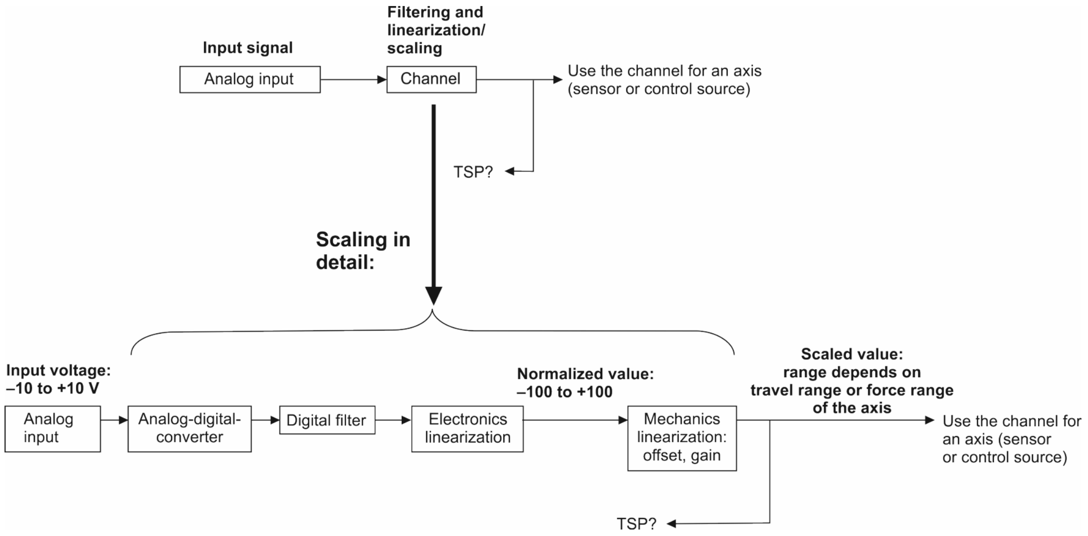


Figure 21: Block diagram for scaling an analog input

Relationship between input voltage and normalized value:

- -10 V correspond to the normalized value -100
- +10 V correspond to the normalized value +100

Relationship between normalized and scaled value:

- Scaled value = offset + gain • normalized value

where the offset and the gain are the corresponding coefficients of the polynomial for mechanics linearization.

In addition to the scaling, the parameters for digital filtering can be set; refer to "Processing Input Signals" (p. 21).

Scaling an analog input

INFORMATION

You can make the settings for scaling the analog input with commands. Alternatively, you have access to the parameters mentioned in the following via the *Axis Definition* and *Sensor Mechanics* parameter groups in the ***Device Parameter Configuration*** window of PIMikroMove.

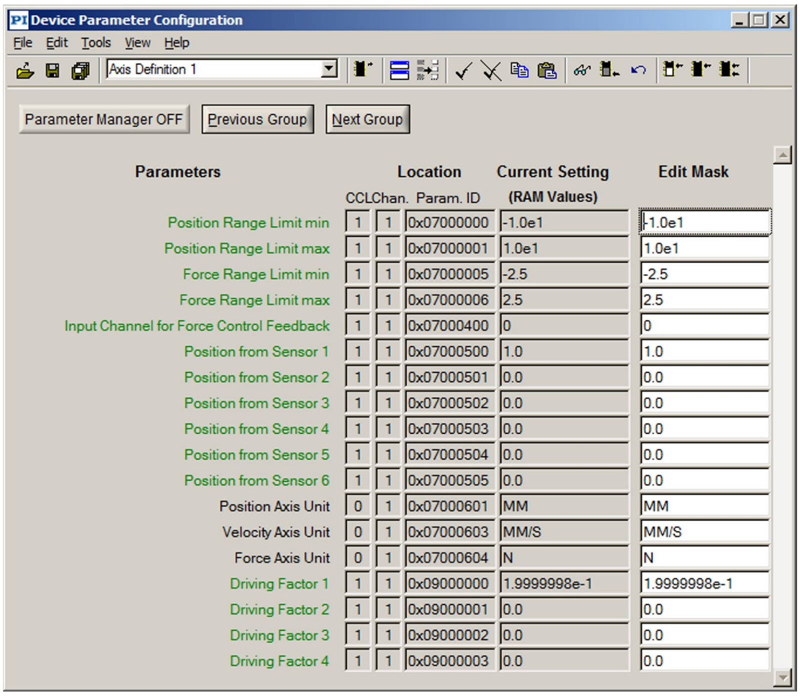


Figure 22: Example: Device Parameter Configuration window in PIMikroMove, the Axis Definition 1 parameter group is shown

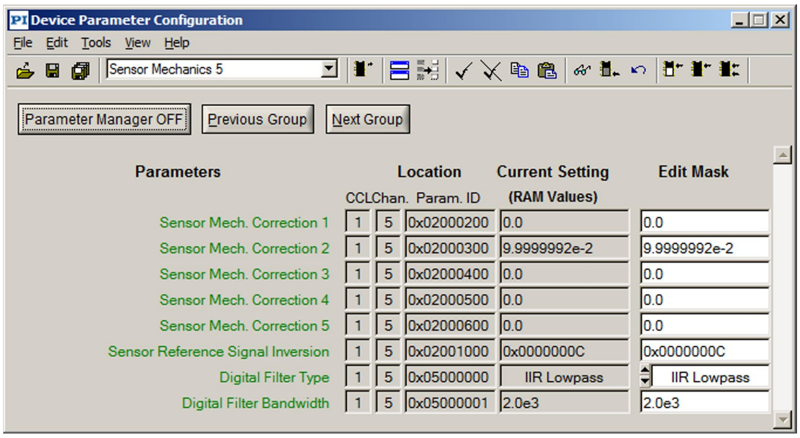


Figure 23: Example: Device Parameter Configuration window in PIMikroMove, the Sensor Mechanics 5 parameter group is shown

When you make the settings in the ***Device Parameter Configuration*** window of PIMikroMove:

- ➢ Read "Device Parameter Configuration" in the PIMikroMove manual.
- ➢ Pay attention to the following instructions when working with commands, but do not determine, change or save the parameter values with the specified commands but instead, in the ***Device Parameter Configuration*** window.
- ➢ When prompted, enter the password *advanced* to go to command level 1.

If you make settings by entering commands (e.g., in PITerminal or in the ***Command entry*** window of PIMikroMove):

1. Determine the travel range or force range limits of the axis for which the analog input is to be used. These limits are referred to as *MinScaledValue* and *MaxScaledValue* in the following.

   If the analog input is to be scaled to the travel range of the axis: *MinScaledValue* is the value of the ***Range Limit min*** parameter (ID 0x07000000), and *MaxScaledValue* is the value of the ***Range Limit max*** parameter (ID 0x07000001).

   If the analog input is to be scaled to the force range of the axis: *MinScaledValue* is the value of the ***Force Range Limit min*** parameter (ID 0x07000005), and *MaxScaledValue* is the value of the ***Force Range Limit max*** parameter (ID 0x07000006).

   – Get the parameter values in the volatile memory with the `TMN? (p. 210)` / `TMX?` (p. 210) or `CMN?` (p. 159) / `CMX?` (p. 161) or `SPA?` (p. 200) commands.

2. Determine the maximum and minimum normalized value corresponding to the input voltage range used in your system. These values are referred to as *MinNormalizedValue* and *MaxNormalizedValue* in the following.

   Examples:

   Input voltage range -10 to +10 V →*MinNormalizedValue* = -100, *MaxNormalizedValue* = +100.

   Input voltage range 0 to +10 V →*MinNormalizedValue* = 0, *MaxNormalizedValue* = +100.

3. Send the `CCL 1 advanced` command to go to command level 1.
4. Scale the analog input by adapting the values for the offset and gain coefficients of the polynomial for mechanics linearization.

   Formulas for calculating the offset and gain coefficients:

   Gain = (*MaxScaledValue* – *MinScaledValue*) / (*MaxNormalizedValue* – *MinNormalizedValue*)

   Offset = *MaxScaledValue* – gain • *MaxNormalizedValue*

   The offset coefficient is the value of the ***Sensor Mech. Correction 1*** parameter (ID 0x02000200). The gain coefficient is the value of the ***Sensor Mech. Correction 2*** parameter (ID 0x02000300).

   a) Calculate the offset and gain coefficients according to the formulas listed above.
   b) Set the ***Sensor Mech. Correction 1*** and ***Sensor Mech. Correction 2*** parameters to the calculated values in the volatile memory with the `SPA` command (p. 199).
5. If linearization is not necessary: Use the `SPA` command to set all other coefficients of the polynomial for mechanics linearization (parameter IDs 0x02000400, 0x02000500, 0x02000600) to the value zero in the volatile memory for the analog input.
6. Optional: Save the parameter settings in the nonvolatile memory of the C-413 with the WPA command (p. 229).

The following examples serve to illustrate the scaling. Hardware properties such as the travel range or the number of input signal channels may differ from your system.

Examples

Input signal channel 5 (analog input 1) is to be scaled to the travel range of axis 1 in the following examples.

The travel range of axis 1 is –20 to +120 µm, i.e.:

- *MinScaledValue* = value of the parameter 0x07000000 = –20 µm
- *MaxScaledValue* = value of parameter 0x07000001 = +120 µm

Example 1

Input voltage range	–10 to +10 V (entire range recommended for maximum resolution)
MinNormalizedValue	–100
MaxNormalizedValue	+100

Gain = (120 – (–20)) / (100 – (–100)) = 0.7

Offset = 120 – 0,7 • 100 = 50

Scaled value = 50 + 0.7 • normalized value

➢ Send the following commands to set the offset and gain coefficients for input signal channel 5:

```
SPA 5 0x02000200 50
```

`SPA 5 0x02000300 0.7`

Example 2

Input voltage range	0 to +10 V (only positive input voltages)
MinNormalizedValue	0
MaxNormalizedValue	+100

Gain = (120 – (–20)) / (100 – 0) = 1.4

Offset = 120 – 1.4 • 100 = –20

Scaled value = –20 + 1.4 • normalized value

➢ Send the following commands to set the offset and gain coefficients for input signal channel 5:

`SPA 5 0x02000200 -20`

`SPA 5 0x02000300 1.4`

Example 3

Positions with a positive sign should correspond to positive input voltages and positions with a negative sign to negative input voltages.

Maximum input voltage:	+10 V
MinNormalizedValue	0
MaxNormalizedValue	+100

The following calculations apply provided that the absolute values of the negative positions are not greater than the positive positions.

Gain = (120 – 0) / (100 – 0) = 1.2

Offset = 120 – 1.2 • 100 = 0

Scaled value = 1.2 • normalized value

➢ Send the following commands to set the offset and gain coefficients for input signal channel 5:

`SPA 5 0x02000200 0`

`SPA 5 0x02000300 1.2`

INFORMATION

These parameter values also apply to an axis with a travel range from 0 to 120 µm if only positive input voltages (0 to +10 V) are applied.

Example 4

As in example 3:
Positions with a positive sign should correspond to positive input voltages and positions with a negative sign to negative input voltages.

Maximum input voltage:	+5 V
MinNormalizedValue	0
MaxNormalizedValue	+50

The following calculations apply provided that the absolute values of the negative positions are not greater than the positive positions.

Gain = (120 – 0) / (50 – 0) = 2.4

Offset = 120 – 2.4 • 50 = 0

Scaled value = 2.4 • normalized value

- Send the following commands to set the offset and gain coefficients for input signal channel 5:

  `SPA 5 0x02000200 0`

  `SPA 5 0x02000300 2.4`

7.4.4 Using an External Sensor

An external position or force sensor can be connected to an analog input. Depending on the sensor type, different parameter settings are required to configure the C-413 for using the external sensor:

- Using an external position sensor (p. 105)
- Using an external force sensor (p. 106)

INFORMATION

You can configure the C-413 with commands for using an external sensor. Alternatively, you have access to the parameters mentioned in the following via the *Axis Definition* parameter group in the ***Device Parameter Configuration*** window of PIMikroMove.

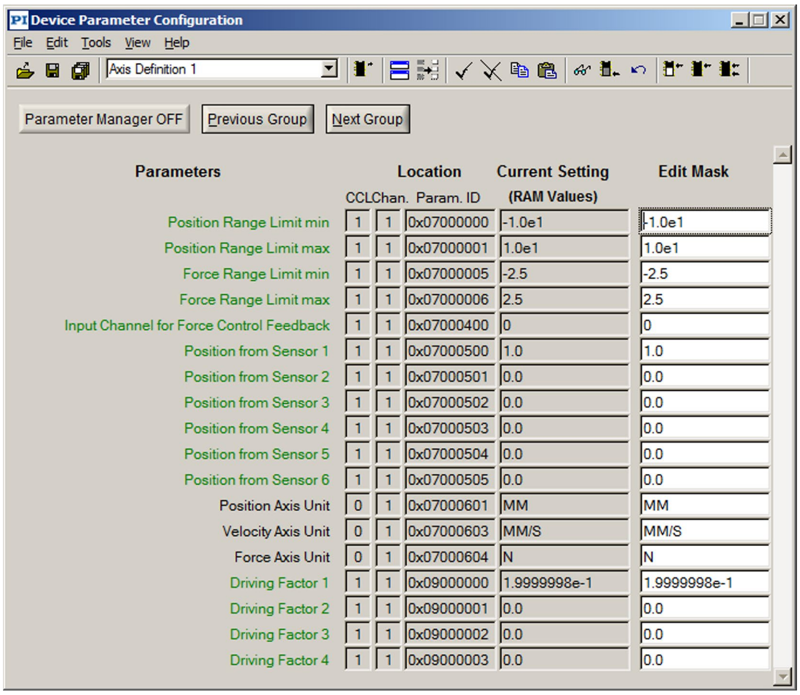


Figure 24: Example: Device Parameter Configuration window in PIMikroMove, the Axis Definition 1 parameter group is shown

Requirements

✓ If you want to use an external position sensor:

- You have connected the position sensor to the **I/O** panel plug (p. 67).
- You have scaled the analog input to the travel range of the axis to be monitored (p. 98).

✓ If you want to use an external force sensor:

- You have connected the force sensor to the **I/O** panel plug (p. 67).

  - You have scaled the analog input to the force range of the axis to be monitored (p. 98).
- ✓ The analog input is **not** connected to an axis for use as a control source (p. 107).

Using an external position sensor

For a position sensor to be used for monitoring an axis, the input matrix (p. 16) must be correspondingly set: The coefficient of the input signal channel belonging to the sensor must have the value 1 for the axis to be monitored.

INFORMATION

In the following cases, the coefficients of an input signal channel must have the value 0 in the input matrix:

- **No** sensor is connected at the input signal channel.
- A position sensor is connected at the input signal channel that is **not** to be used for monitoring an axis.
- A force sensor is connected at the input signal channel (p. 106).
- The input signal channel is used as the control source (p. 107).

The following instructions for setting the input matrix use an example for better comprehension: The position of axis 1 is only to be monitored by the external sensor on input signal channel 5 (analog input 1).

When you make the settings in the ***Device Parameter Configuration*** window of PIMikroMove:

- ➢ Read "Device Parameter Configuration" in the PIMikroMove manual.
- ➢ Pay attention to the following instructions when working with commands, but do not determine, change or save the parameter values with the specified commands but instead, in the ***Device Parameter Configuration*** window.
- ➢ When prompted, enter the password *advanced* to go to command level 1.

If you make settings by entering commands (e.g., in PITerminal or in the ***Command entry*** window of PIMikroMove):

1. Send the `CCL 1 advanced` command to go to command level 1.
2. Set the coefficient of input signal channel 5 to the value 1 for axis 1 in the input matrix:
   - Send the `SPA 1 0x07000504 1` command.
3. Set the coefficients of all other input signal channels to the value zero for axis 1 in the input matrix.
   - Send the command `SPA 1 0x07000500 0 1 0x07000501 0 1 0x07000502 0`.
   - Send the command `SPA 1 0x07000503 0 1 0x07000505 0 1`.

   The division into two commands is necessary because a maximum of 12 arguments are permitted per command.

4. Optional: Check whether the position sensor is actually used for monitoring axis 1:
   - Send the `POS? 1` command to get the current position of axis 1.
   - Send the `TSP? 5` command to get the current position response of input signal channel 5.

   When the position sensor at input signal channel 5 is used for monitoring axis 1, the values in both responses are identical.
5. Optional: Save the parameter settings in the nonvolatile memory of the C-413 with the `WPA` command (p. 229).

Using an external force sensor

For a force sensor to be used to monitor an axis, the force sensor must be directly allocated to the axis via the ***Input Channel For Force Feedback*** parameter (ID 0x07000400).

INFORMATION

When a force sensor is connected to an input signal channel, the coefficients of the input matrix (p. 16) must have the value zero for this input signal channel.

The following instructions for allocating the force sensor use an example for better comprehension: The force applied by axis 2 is to be monitored by the external sensor on input signal channel 6 (analog input 2).

When you make the settings in the ***Device Parameter Configuration*** window of PIMikroMove:

- ➢ Read "Device Parameter Configuration" in the PIMikroMove manual.
- ➢ Pay attention to the following instructions when working with commands, but do not determine, change or save the parameter values with the specified commands but instead, in the ***Device Parameter Configuration*** window.
- ➢ When prompted, enter the password *advanced* to go to command level 1.

If you make settings by entering commands (e.g., in PITerminal or in the ***Command entry*** window of PIMikroMove):

1. Send the `CCL 1 advanced` command to go to command level 1.
2. Allocate the force sensor at input signal channel 6 to axis 2:
   - Send the command `SPA 2 0x07000400 6`.
3. Set the coefficients of input signal channel 6 to the value zero for all axes in the input matrix.
   - Send the command `SPA 1 0x07000505 0 2 0x07000505 0`.
4. Optional: Check whether the force sensor is actually used for monitoring axis 2:
   a) Make sure that the control variable for axis 2 is the force (e.g., by selecting the control mode with the `CMO` command (p. 159)).
   b) Send the `CAV? 2` command to get the current force of axis 2.

– Send the `TSP? 6` command to get the current force response of input signal channel 6.

When the force sensor at input signal channel 6 is used for monitoring axis 2, the values in both responses are identical.

5. Optional: Save the parameter settings in the nonvolatile memory of the C-413 with the `WPA` command (p. 229).

7.4.5 Using as a Control Source

The signal at the analog input can be used as the control source for an axis. The following steps are necessary to use an analog input as a control source:

- Connecting the input signal channel and axis
- Optional: Setting the offset for the axis

Depending on the servo mode (p. 22), the analog input specifies absolute target values or control values.

When an analog input is used as the control source for an axis, it is not permitted to switch the servo mode on or off or to change the control mode (p. 28) for the axis.

When the corresponding setting is written to the nonvolatile memory, the axis can be controlled with the analog input signal directly after the C-413 is switched on (**no** PC necessary).

INFORMATION

You can configure the C-413 with commands for using the analog input as a control source. Alternatively, you have access to the parameters mentioned in the following via the *Target Manipulation* parameter groups in the ***Device Parameter Configuration*** window of PIMikroMove.

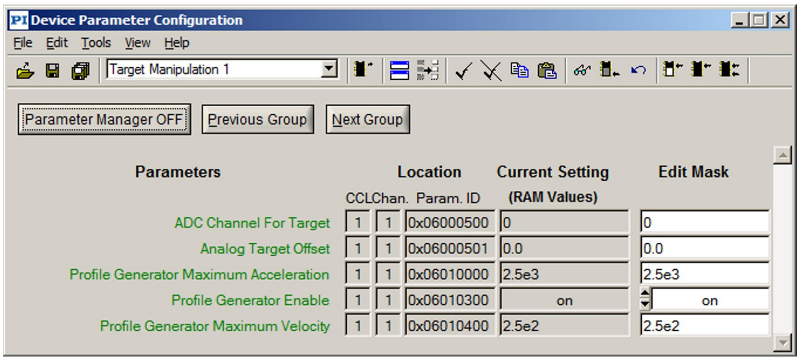


Figure 25: Example: Device Parameter Configuration window in PIMikroMove, the Target Manipulation 1 parameter group is shown

Requirements

✓ You have connected a suitable signal source to the **I/O** panel plug (p. 67).

✓ You have scaled the analog input to the travel range or force range of the axis to be controlled (p. 98).

✓ The analog input is **not** used for an external sensor (p. 103).

Connecting the input signal channel and axis

The following instructions for connecting an analog input signal to an axis use an example for better comprehension: Axis 1 is to be commanded by an analog signal at input signal channel 5 (analog input 1).

When you make the settings in the ***Device Parameter Configuration*** window of PIMikroMove:

➢ Read "Device Parameter Configuration" in the PIMikroMove manual.

➢ Pay attention to the following instructions when working with commands, but do not determine, change or save the parameter values with the specified commands but instead, in the ***Device Parameter Configuration*** window.

➢ When prompted, enter the password *advanced* to go to command level 1.

If you make settings by entering commands (e.g., in PITerminal or in the ***Command entry*** window of PIMikroMove):

1. Send the `CCL 1 advanced` command to go to command level 1.
2. Connect input signal channel 5 with axis 1:
   - Send the `SPA 1 0x06000500 5` command.

   The allocation of the input signal channel to the axis activates the analog input as a control source. The analog input now overwrites the target or control values that are specified by motion commands or by the wave generator; refer also to "Generating Control Values" (p. 24).
3. Optional: Check whether the analog input is actually used as a control source for axis 1:
   - Change the analog signal at input signal channel 5 and observe the behavior of axis 1 at the same time.
4. Optional: Save the parameter settings in the nonvolatile memory of the C-413 with the `WPA` command (p. 229).
5. If necessary: Stop the axis with the `STP` (p. 204) or `#24` (p. 150) command.

   Stopping the axis with `STP` or `#24` terminates the connection between the input signal channel and axis (parameter 0x06000500 is set to the value zero for the axis).

   If you want to command the axis via the analog input again:
   - Connect the input signal channel with the axis again; see step 2.

Optional: Setting the offset for the axis

The offset for the axis is specified by the ***Analog Target Offset*** parameter (ID 0x06000501). When an input signal channel is connected to the axis as a control source, the offset is added to the current value of the input signal channel.

When you make the settings in the ***Device Parameter Configuration*** window of PIMikroMove:

➢ Read "Device Parameter Configuration" in the PIMikroMove manual.

- Pay attention to the following instructions, but do not determine, change or save the parameter values with the specified commands but instead, in the ***Device Parameter Configuration*** window.
- When prompted, enter the password *advanced* to go to command level 1.

If you make settings by entering commands (e.g., in PITerminal or in the ***Command entry*** window of PIMikroMove):

- Determine the current offset for the axis by querying the value of the ***Analog Target Offset*** parameter in the volatile memory:
  - Send the `AOS?` command (p. 153).

  \- or -

  - Query the value of the parameter with the `SPA?` command (p. 200).
- Determine the default setting for the axis offset by querying the value of the ***Analog Target Offset*** parameter in the nonvolatile memory with the `SEP?` command (p. 198).
- Set the axis offset by changing the value of the ***Analog Target Offset*** parameter:
  - Write the new offset to the volatile memory with the `AOS` command (p. 151).

  \- or -

  a) Send the `CCL 1 advanced` command to go to command level 1.

  b) Change the offset in the volatile memory with the `SPA` command (p. 199) or in the nonvolatile memory with the `SEP` command (p. 197).

  If you want to write the value of the parameter from the volatile to the nonvolatile memory of the C-413:

  c) Send the `CCL 1 advanced` command to go to command level 1.

  d) Save the parameter value with the `WPA` command (p. 229).

7.4.6 Deactivating an Analog Input

To prevent malfunctions, you have to deactivate an analog input that is neither used for an external sensor nor as a control input.

The following instructions for deactivating an analog input use an example for better comprehension: Analog input 1 (input signal channel 5) is to be deactivated.

When you make the settings in the ***Device Parameter Configuration*** window of PIMikroMove:

- Read "Device Parameter Configuration" in the PIMikroMove manual.
- Pay attention to the following instructions when working with commands, but do not determine, change or save the parameter values with the specified commands but instead, in the ***Device Parameter Configuration*** window.
- When prompted, enter the password *advanced* to go to command level 1.

If you make settings by entering commands (e. g. in PITerminal or in the ***Command entry*** window of PIMikroMove):

1. Send the `CCL 1 advanced` command to go to command level 1.

2. Send the command `SPA 1 0x07000504 0 2 0x07000504 0` to deactivate input signal channel 5 for axes 1 and 2 in the volatile memory (***Position From Sensor 5*** parameter = 0).
3. Send the command `SPA? 1 0x06000500 2 0x06000500` to check whether input signal channel 5 is still connected to an axis as a control source.

   The response provides the values of the ***ADC Channel For Target*** parameter for the individual axes.

   Example:

   `1 0x06000500=5`

   `2 0x06000500=6`

   According to the above response, axis 1 is commanded by input signal channel 5 and axis 2 by input signal channel 6.
4. If the input signal channel to be deactivated is still connected with an axis as a control source, disconnect the connection by setting the ***ADC Channel For Target*** parameter. In the example, send the command `SPA 1 0x06000500 0` to disconnect.
5. Optional: Save the parameter settings in the nonvolatile memory of the C-413 with the `WPA` command (p. 229).

7.5 Analog Output Signals

This section describes how to use the analog outputs that are present on the **I/O** panel plug (p. 295) of the C-413.2GA and C-413.20A models.

7.5.1 Utilization Types for Analog Outputs

The analog outputs on the **I/O** panel plug can be addressed in the firmware of the C-413 as output signal channels 3 and 4 and are intended for the following types of use:

- Controlling an external motor driver (p. 114)
- Monitoring the position, force or velocity of an axis (p. 116)

The following figure shows the block diagram for the types of use.

Possible uses of an analog output (output signal channel 3 or 4)

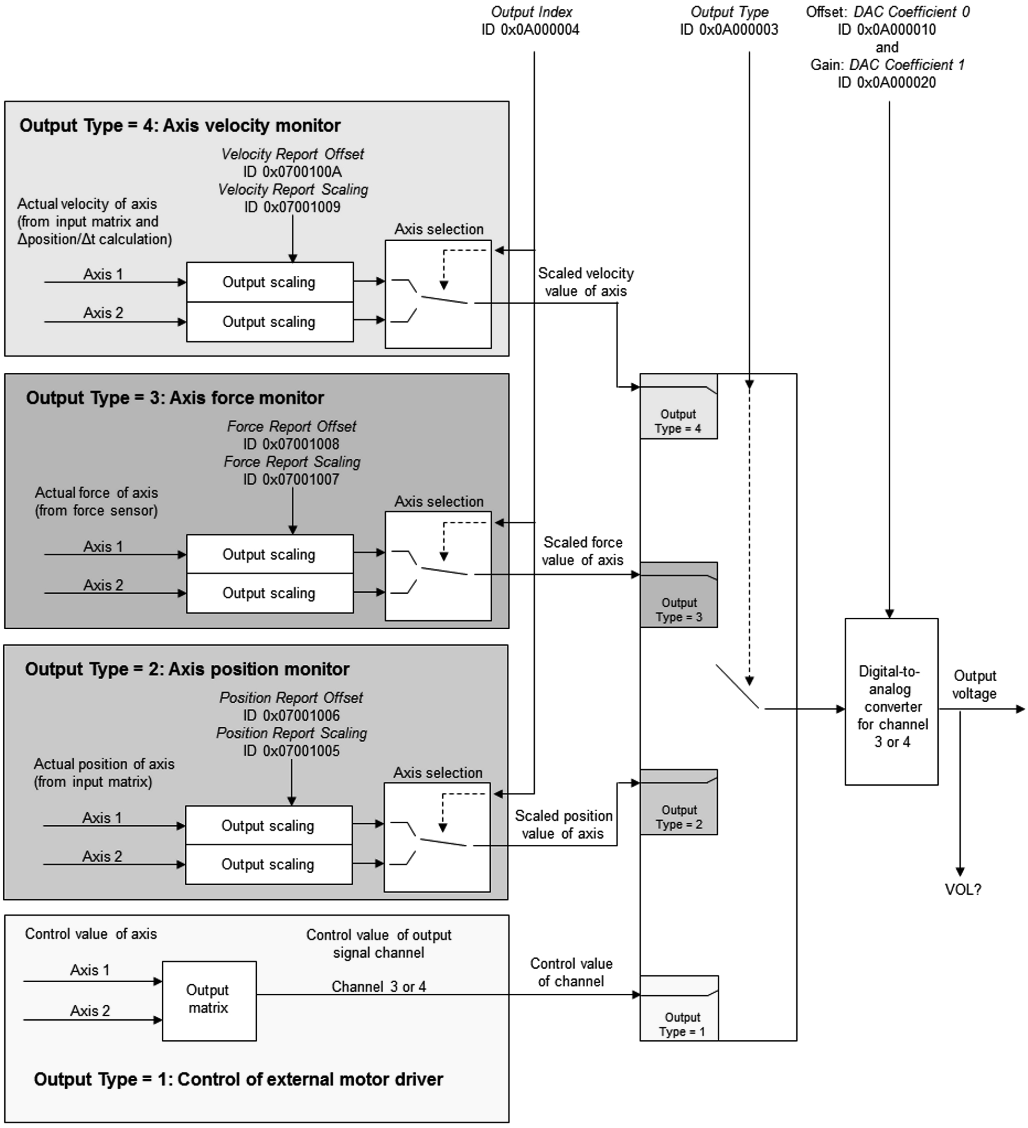


Figure 26: Block diagram for using an analog output (output signal channel 3 or 4)

INFORMATION

➢ Use the complete voltage range of the analog output (–10 to +10 V) to achieve maximum resolution.

If necessary, the digital/analog converter of the analog output can be adjusted (p. 118).

7.5.2 Commands and Parameters for Analog Outputs

Commands

The following commands are available for using an analog output:

Command	Syntax	Function
TPC?	TPC?	Gets the total number of output signal channels (value of the ***Number Of Output Channels*** parameter)
VOL?	VOL? [{<OutputSignalID>}]	Gets the current value of the output signal channel (for output signal channels 3 and 4: Output voltage in V).

Parameters

The following parameters are available for configuring an analog output:

Parameter	Description and Possible Values
Position Report Scaling ID 0x07001005	Gain for output scaling of the axis position Necessary for scaling the axis position to the analog output, refer to "Using to Monitor the Position, Force or Velocity of an Axis" (p. 116) for further information.
Position Report Offset ID 0x07001006	Offset for output scaling of the axis position Necessary for scaling the axis position to the analog output, refer to "Using to Monitor the Position, Force or Velocity of an Axis" (p. 116) for further information.
Force Report Scaling ID 0x07001007	Gain for output scaling of the force of the axis Necessary for scaling the force of the axis to the analog output, refer to "Using to Monitor the Position, Force or Velocity of an Axis" (p. 116) for further information.
Force Report Offset ID 0x07001008	Offset for output scaling of the force of the axis Necessary for scaling the force of the axis to the analog output, refer to "Using to Monitor the Position, Force or Velocity of an Axis" (p. 116) for further information.
Velocity Report Scaling ID 0x07001009	Gain for output scaling of the velocity of the axis Necessary for scaling the velocity of the axis to the analog output, refer to "Using to Monitor the Position, Force or Velocity of an Axis" (p. 116) for further information.
Velocity Report Offset ID 0x0700100A	Offset for output scaling of the of the velocity of the axis Necessary for scaling the velocity of the axis to the analog output, refer to "Using to Monitor the Position, Force or Velocity of an Axis" (p. 116) for further information.
Driving Factor 1 ID 0x09000000 ***Driving Factor 2*** ID 0x09000001	Allocation of output signal channels to axes The parameters indicate the coefficients of the output matrix that is intended for converting the control values of the axes into the control values of the output signal channels. Refer to "Allocating Axes to Channels" (p. 16) and "Using as a Control

Parameter	Description and Possible Values
Driving Factor 3 ID 0x09000002	Signal for an External Motor Driver" (p. 114).
Driving Factor 4 ID 0x09000003	
Output Type ID 0x0A000003	Selection of the type of use of the output signal channel 1 = Control signal for external motor driver 2 = Monitor of the position of the axis 3 = Monitor of the force of the axis 4 = Monitor of the velocity of the axis This parameter is preset to "control signal" (value 1) for output signal channels 1 and 2 and write-protected. Refer to "Using as a Control Signal for an External Motor Driver" (p. 114) and "Using to Monitor the Position, Force or Velocity of an Axis" (p. 116) for further information.
Output Index ID 0x0A000004	Identifier of the axis to be monitored The use of the parameter depends on the value of the ***Output Type*** parameter (ID 0x0A000003): ***Output Type*** = 1: ***Output Index*** is not used. ***Output Type*** = 2 or 3 or 4: ***Output Index*** indicates the identifier of the axis whose position, force or velocity is to be output at the output signal channel. Refer to "Using to Monitor the Position, the Force or Velocity of an Axis" (p. 116) for further information.
DAC Coefficient 0 ID 0x0A000010	Offset for the digital/analog converter Adjusts the measured output value of the output signal channel to the response to the `VOL?` command. Refer to "Adjusting the Digital/Analog Converter of the Analog Output" (p. 118) for further information.
DAC Coefficient 1 ID 0x0A000020	Gain for the digital/analog converter Adjusts the measured output value of the output signal channel to the response to the `VOL?` command. Refer to "Adjusting the Digital/Analog Converter of the Analog Output" (p. 118) for further information.
Number of Output Channels ID 0x0E000B01	Total number of output signal channels The number of available analog outputs is the difference between the total number of output signal channels and the number of motor driver outputs (***Number Of Driver Channels***, ID 0x0E000B04).
Number of Driver Channels ID 0x0E000B04	Number of motor driver outputs Output signal channels that are only intended for motor drivers; outputs via the **Motor & Sensor** sockets (p. 294).

7.5.3 Using as a Control Signal for an External Motor Driver

An external motor driver can be controlled via an analog output of the C-413 (output signal channel 3 or 4). Required settings for the corresponding output signal channel:

- Set the output matrix for calculating the control value of the channel
- Select "Control signal for external motor driver" as the type of use

If necessary, the digital/analog converter of the analog output can be adjusted (p. 118).

INFORMATION

The control signal that is output by the analog output must be scaled to the connected external motor driver. The scaling takes place via the coefficients of the corresponding output signal channel in the output matrix (p. 19).

The coefficients of the output matrix are floating-point numbers in the value range from 0 to 1:

0 = Control value of the axis is not output at the channel

1 = Control value of the channel corresponds 1:1 to the control value of the axis (maximum scaling)

INFORMATION

You can configure the C-413 with commands for using an analog output for controlling an external motor driver. Alternatively, in the ***Device Parameter Configuration*** window of PIMikroMove, you have access to the parameters in the *Axis Definition* and *DAC* parameter groups mentioned in the following.

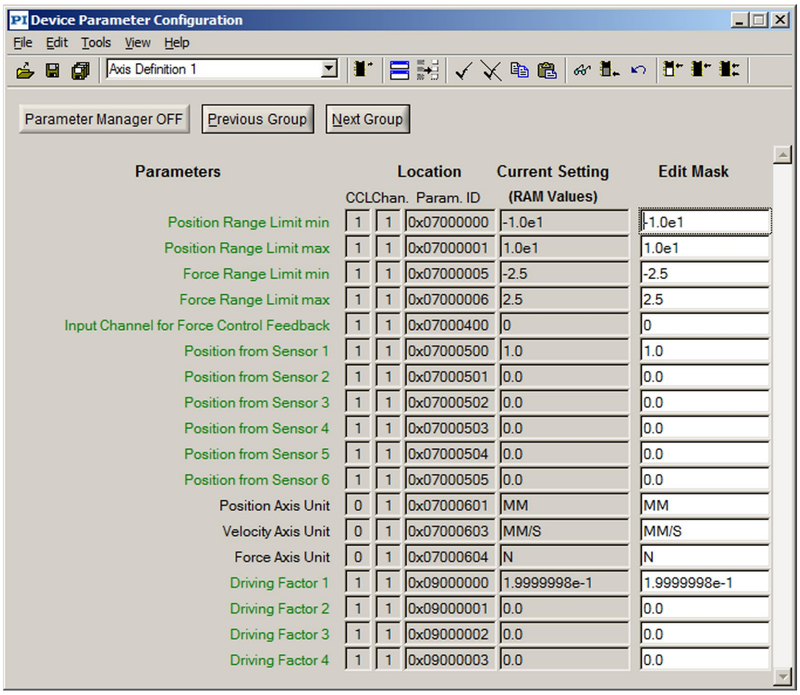


Figure 27: Example: Device Parameter Configuration window in PIMikroMove, the Axis Definition 1 parameter group is shown

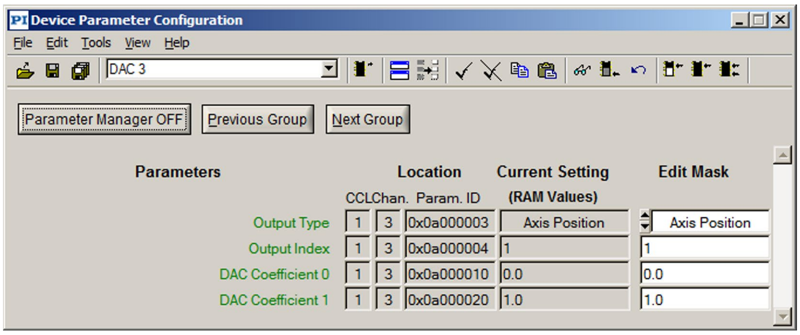


Figure 28: Example: Device Parameter Configuration window in PIMikroMove, the DAC 3 parameter group is shown

Requirements

✓ You have connected a suitable motor driver to the **I/O** panel plug (p. 67).

Using an analog output as a control signal for an external motor driver

The following instructions use an example for better comprehension: The control value of axis 1 is to be output 1:1 as an analog control signal at output signal channel 3 (analog output 1).

When you make the settings in the ***Device Parameter Configuration*** window of PIMikroMove:

➢ Read "Device Parameter Configuration" in the PIMikroMove manual.

➢ Pay attention to the the following instructions when working with commands, but do not determine, change or save the parameter values with the specified commands but instead, in the ***Device Parameter Configuration*** window.

➢ When prompted, enter the password *advanced* to go to command level 1.

If you make settings by entering commands (e.g., in PITerminal or in the ***Command entry*** window of PIMikroMove):

1. Send the `CCL 1 advanced` command to go to command level 1.
2. Set the output matrix for calculating the control value of output signal channel 3 (see also "Output matrix" (p. 19)):
   – Send the command `SPA 1 0x09000002 1.0` to set the coefficient of output signal channel 3 for axis 1 to the value 1 (1:1 output of the control value of axis 1).
   – Send the command `SPA 2 0x09000002 0.0` to set the coefficient of output signal channel 3 for axis 2 to the value zero (no output of the control value of axis 2).

   Both commands can also be combined in one command line.
3. Select "Control signal for external motor driver" as the type of use for output signal channel 3:
   – Send the command `SPA 3 0x0A000003 1`.
4. Optional: Save the parameter settings in the nonvolatile memory of the C-413 with the `WPA` command (p. 229).

7.5.4 Using to Monitor the Position, the Force or Velocity of an Axis

The position, force or velocity of an axis can be output via an analog output of the C-413 (output signal channel 3 or 4). Required settings:

- Scale the position, force or velocity of the axis to the analog output; for an example see the figure below
- For the corresponding output signal channel, select the type of use "Monitor" in the desired size and set the identifier of the axis for which the desired size is to be output

If necessary, the digital/analog converter of the analog output can be adjusted (p. 118).

The following block diagram shows the scaling to an analog output using the position of the axis as an example. The scaling of the force and velocity of the axis is done in the same way using the corresponding parameters for offset and gain.

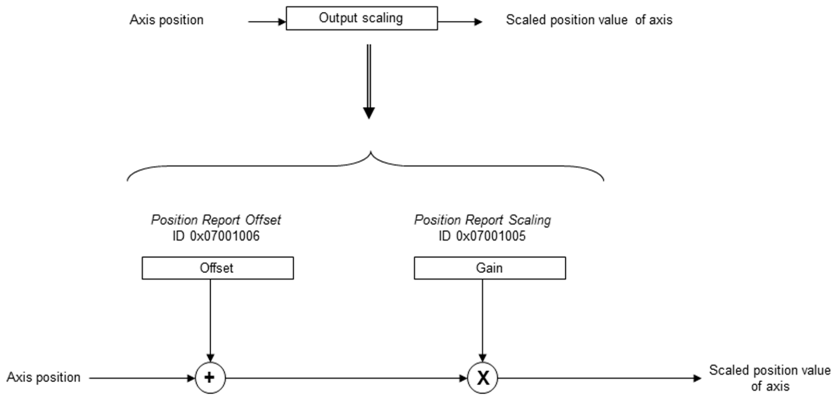


Figure 29: Block diagram for scaling the position of an axis to an analog output

INFORMATION

You can configure the C-413 with commands for using an analog output as a monitor. Alternatively, in the ***Device Parameter Configuration*** window of PIMikroMove, you have access to the parameters in the *Servo* and *DAC* parameter groups mentioned in the following.

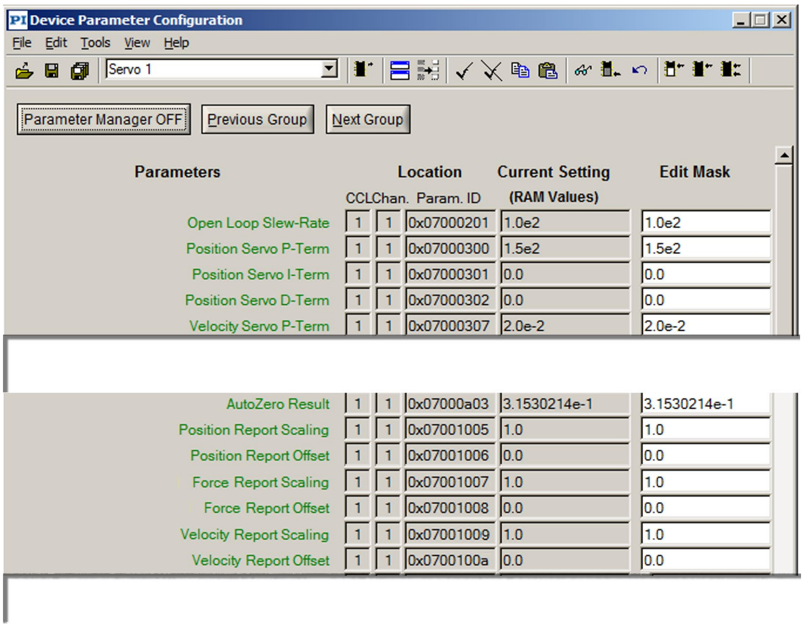


Figure 30: Example: Excerpts from the Device Parameter Configuration window in PIMikroMove, the Servo 1 parameter group is shown

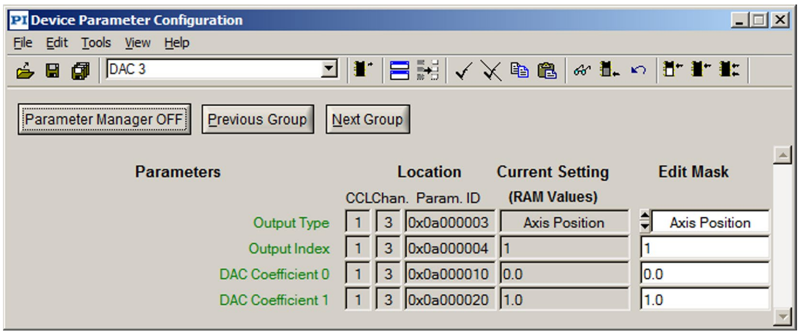


Figure 31: Example: Device Parameter Configuration window in PIMikroMove, the DAC 3 parameter group is shown

Requirements

✓ You have connected a suitable measuring device at the **I/O** panel plug (p. 67).

Using an analog output to monitor the position, force or velocity

The following instructions use an example for better comprehension: The position of axis 2 is to be output at analog output 2 (output signal channel 4).

Travel range of axis 2: -3 mm to 5 mm

Range to be used of analog output 2: -10 V to 10 V

When you make the settings in the ***Device Parameter Configuration*** window of PIMikroMove:

- Read "Device Parameter Configuration" in the PIMikroMove manual.
- Pay attention to the following instructions when working with commands, but do not determine, change or save the parameter values with the specified commands but instead, in the ***Device Parameter Configuration*** window.
- When prompted, enter the password *advanced* to go to command level 1.

If you make settings by entering commands (e.g., in PITerminal or in the ***Command entry*** window of PIMikroMove):

1. Send the `CCL 1 advanced` command to go to command level 1.
2. Scale the size that is to be output for the axis to the analog output by specifying the offset and gain.

   The parameters to be set depend on the size to be output:

   – Position: 0x07001006 (offset), 0x07001005 (gain)

   – Force: 0x07001008 (offset), 0x07001007 (gain)

   – Velocity: 0x0700100A (offset), 0x07001009 (gain)

   In the example, an offset of -1 and a gain of 2.5 are necessary to represent the travel range of axis 2 on the analog output range to be used.

   – Send the command `SPA 2 0x07001006 -1 2 0x07001005 2.5` to set the offset and gain for the position of axis 2.
3. Select the type of use for the output signal channel by setting the parameter 0x0A000003 to the corresponding value.

   Parameter values for using as an axis monitor:

   – Position: 2

   – Force: 3

   – Velocity: 4

   In the example, the type of use "Monitor of the position of the axis" is selected for analog output 2 (output signal channel 4).

   – Send the command `SPA 4 0x0A000003 2`.
4. Set the identifier of the axis for which the size selected in step 3 is to be output (here axis 2) for the output signal channel (here channel 4):

   – Send the command `SPA 4 0x0A000004 2`.
5. Optional: Save the parameter settings in the nonvolatile memory of the C-413 with the `WPA` command (p. 229).

7.5.5 Adjusting the Digital/Analog Converter of the Analog Output

It is necessary to adjust the digital/analog converter of an analog output when the measured output value deviates from the response to the `VOL?` command for the corresponding output signal channel. During the adjustment, the offset and gain are set for the digital/analog converter.

INFORMATION

You can make the settings for adjusting the digital/analog converter with commands. Alternatively, you have access to the parameters mentioned in the following via the *DAC* parameter group in the ***Device Parameter Configuration*** window of PIMikroMove. In the main window of PIMikroMove you can read the current value of the output signal channel (response to `VOL?`) on the ***Output channels*** card.

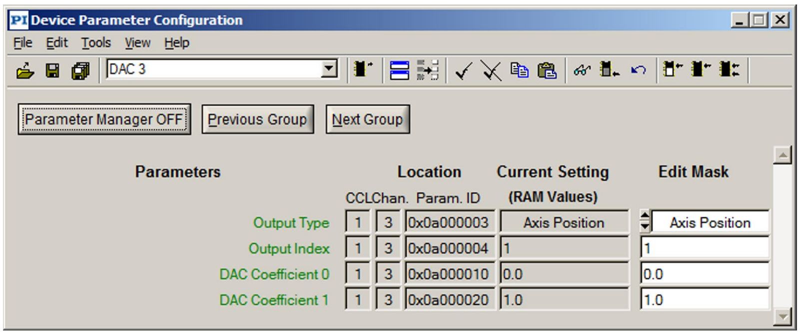


Figure 32: Example: Device Parameter Configuration window in PIMikroMove, the DAC 3 parameter group is shown

Requirements

✓ You have connected a suitable measuring device at the **I/O** panel plug (p. 67).

Adjusting the digital/analog converter of the analog output

The following instructions for adjusting the digital/analog converter of an analog output use an example for better comprehension: The digital/analog converter of analog output 1 (output signal channel 3) is to be adjusted.

When you make the settings in the ***Device Parameter Configuration*** window of PIMikroMove:

➢ Read "Device Parameter Configuration" in the PIMikroMove manual.

➢ Pay attention to the following instructions when working with commands, but do not determine, change or save the parameter values with the specified commands but instead, in the ***Device Parameter Configuration*** window.

➢ When prompted, enter the password *advanced* to go to command level 1.

If you make settings by entering commands (e.g., in PITerminal or in the ***Command entry*** window of PIMikroMove):

1. Send the `CCL 1 advanced` command to go to command level 1.

2. Send the `VOL? 3` command to get the current value of output signal channel 3 (analog output 1).
3. Determine the actual output value at analog output 1 by measuring with the connected measuring device.
4. If the queried value deviates from the measured value:
   – Send the command `SPA 3 0x0A000010 Offset`, where *Offset* specifies the offset value for the digital/analog converter of output signal channel 3.
   – Send the command `SPA 3 0x0A000020 Gain`, where *Gain* specifies the gain value for the digital/analog converter of output signal channel 3.
5. Repeat steps 2, 3 and 4 in this order until the queried and the measured value match.
6. Optional: Save the parameter settings in the nonvolatile memory of the C-413 with the `WPA` command (p. 229).

7.6 Wave Generator

7.6.1 Functionality of the Wave Generator

The two wave generators of the C-413 are intended to be used as control sources for the axis motion; refer also to "Generating Control Values" (p. 24). The wave generators are permanently assigned to the axes of the C-413: Wave generator 1 to axis 1, wave generator 2 to axis 2.

A wave generator outputs the target or control values for the axis motion on the basis of defined waveforms, depending on the servo mode. The wave generator output is especially suited to dynamic applications with periodic axis motions.

The following block diagram shows the integration of a wave generator in the C-413.

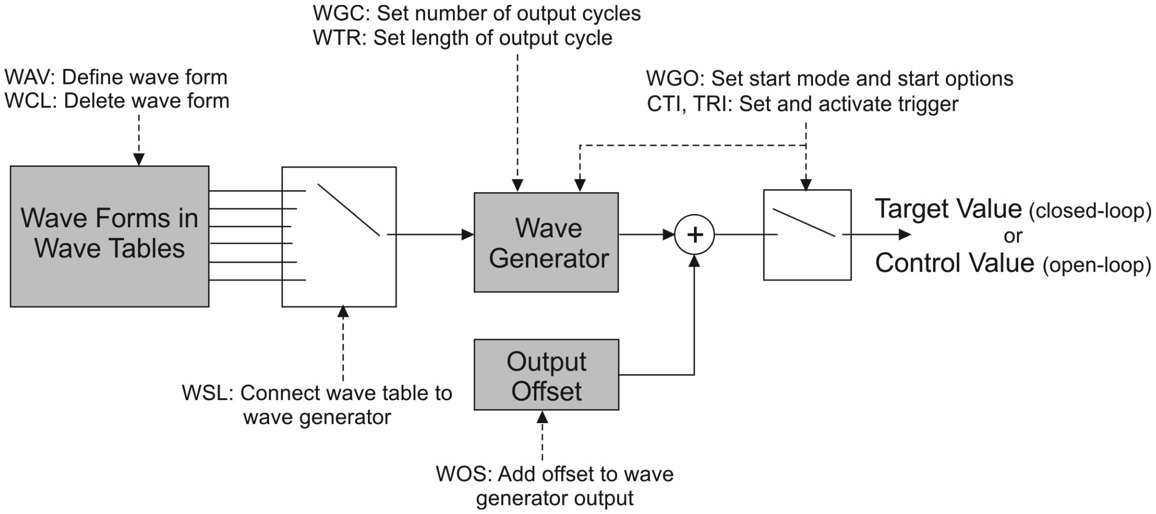


Figure 33: Block diagram of a wave generator

Wave tables

Waveforms can be defined and temporarily stored in eight wave tables in the volatile memory of the C-413 (p. 124). Each wave table contains the data of one waveform. The total number of points of the wave tables is 4096.

The wave tables can be assigned to the wave generators and thus the axes as desired (p. 133). A wave table can be used by several wave generators at the same time.

Change in the output cycles

The number of output cycles (p. 134) and the output rate (p. 134) of the wave generator can be set with commands and parameters. An offset can also be added to the output waveform (p. 133).

Triggering

Programmable trigger inputs allow the wave generator output to be started and interrupted with digital input signals (p. 90).

Wave generator in closed-loop and open-loop operation

Depending on the servo mode (p. 22), the wave generator outputs absolute target values or control values.

When the wave generator is running for the corresponding axis, it is not permitted to switch the servo mode on or off or to change the control mode (p. 28).

INFORMATION

It is recommended to use PIMikroMove for working with the wave generator.

7.6.2 Commands and Parameters for the Wave Generator

Commands

The following commands are available for using the wave generator:

Command	Syntax	Function
`GWD?`	GWD? [<StartPoint> <NumberOfPoints> [{<WaveTableID>}]]	Queries the content of the wave tables (i.e., the waveforms).
`TWG?`	TWG?	Queries the number of wave generators (= number of axes).
`WAV`	WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>	Defines the waveform.
`WAV?`	WAV? [{<WaveTableID> <WaveParameterID>}]	Queries the current length of the wave tables (number of points).
`WCL`	WCL {<WaveTableID>}	Deletes the contents of the wave tables.

Command	Syntax	Function
WGC	WGC {<WaveGenID> <Cycles>}	Sets the number of output cycles.
WGC?	WGC? [{<WaveGenID>}]	Queries the number of output cycles.
WGO	WGO {<WaveGenID> <StartMode>}	Sets the mode and options for starting and stopping the wave generator output. When the start mode "Start by external trigger signal" is set: The trigger configuration is set with CTI and enabled with TRI.
WGO?	WGO? [{<WaveGenID>}]	Queries the start mode and start option(s) last commanded for the wave generator.
WGR	WGR	Starts the data recording again while the wave generator is running.
WOS	WOS {<WaveGenID> <Offset>}	Writes the offset to be added to the wave generator output into the volatile memory.
WOS?	WOS? [{<WaveGenID>}]	Queries the offset to be added to the wave generator output from the volatile memory.
WSL	WSL {<WaveGenID> <WaveTableID>}	Establishes the connection between the wave table and the wave generator.
WSL?	WSL? [{<WaveGenID>}]	Queries the connection between the wave table and the wave generator.
WTR	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}	Sets the table rate of the wave generator (therefore influencing the duration of an output cycle).
WTR?	WTR? [{<WaveGenID>}]	Queries the table rate of the wave generator.
#9	#9	Queries the current activation state of the wave generator.

Parameters

The following parameters are available for configuring the wave generator:

Parameters	Description and Possible Values
Maximum Number Of Wave Points (ID 0x13000004)	Total number of available points for waveforms The wave tables of the C-413 have a total of §§§ mm points. The available points are distributed among the wave tables when waveforms are defined with the WAV command (p. 217). This parameter is write-protected.
Wave Generator Table Rate (ID 0x13000109)	Wave generator output rate Integer value > 0 The individual output cycles of the waveform can be lengthened with the value of the parameter. The value of the parameter can also be set in the volatile memory with

Parameters	Description and Possible Values
	the `WTR` command (p. 231). Refer to "Configuring a Wave Generator" (p. 132) for more information.
Number of Wave Tables (ID 0x1300010A)	Number of wave tables for saving waveforms The C-413 has 8 wave tables. This parameter is write-protected.
Wave Offset (ID 0x1300010B)	Output offset for the wave generator The current wave generator output is generated as follows: Generator output = output offset + current wave value The value of the parameter can also be set in the volatile memory with the `WOS` command (p. 227). Refer to "Configuring a Wave Generator" (p. 132) for more information.

INFORMATION

The following settings for using the wave generator can only be changed in the volatile memory of the C-413 and are lost when the C-413 is switched off or rebooted:

- Wave table content: `WAV`
- Assignment of wave tables to wave generators: `WSL`
- Trigger configuration for wave generator output: `CTI` and `TRI`
- Number of output cycles of the wave generator: `WGC`

The following settings can be stored in the nonvolatile memory of the C-413 with the `WPA` command:

- Output offset for the wave generator: `WOS` /***Wave Offset*** parameter
- Wave generator table rate: `WTR` / ***Wave Generator Table Rate*** parameter

7.6.3 Defining the Waveform

Waveforms are defined with the following steps:

- Optional: Getting information on wave tables (p. 124)
- Creating a waveform in a wave table (p. 124)
- Optional: Deleting the wave table content (p. 125)

This manual contains examples for creating waveforms (p. 125).

INFORMATION

The wave table content (= defined waveforms) is only present in the volatile memory of the C-413 and is lost when the C-413 is switched off or rebooted.

Optional: Getting information on wave tables

- Send the `SPA? 1 0x13000004` command to get the total number of points that the C-413 provides for defining waveforms in wave tables.
- Send the `SPA? 1 0x1300010A` command to get the number of wave tables available in the C-413.
- Get the current number of already defined waveform points for the wave tables with the `WAV?` command (p. 223).
- Get the current contents of the wave tables (= already defined waveforms) with the `GWD?` command (p. 180).

  The response contains the wave table content in the GCS array format (refer to the separate manual for GCS array, SM 146E).

  The response does not contain the output offset defined with the `WOS` command (p. 227), which is not added until the waveforms are output (p. 133).

Creating a waveform in a wave table

1. Make sure that the selected wave table is **not** connected to a wave generator for which the output has been started. Refer to "Configuring a Wave Generator" (p. 132) and "Stopping the Wave Generator Output" (p. 137) for details.
2. Create the waveform in the selected wave table from individual segments with the `WAV` command (p. 217) (WAV + max. 12 arguments). Supported curve types:

   "PNT" (user-defined curve)

   "SIN_P" (inverted cosine curve)

   "RAMP" (ramp curve)

   "LIN" (curve in the form of a single scan line)

   The waveform is written in the selected wave table in the volatile memory. Refer to "Examples for creating waveforms" (p. 125) for details.

INFORMATION

In closed-loop operation, the interpretation of the values of the waveform points depends on the selected control mode. In open-loop operation, the values of the waveform points correspond to the force to be applied in N (refer to "Generating Control Values" (p. 24) and "Control Modes and Control Variables" (p. 28) for further information).

When a waveform is defined with `WAV` (p. 217), the resulting target values (closed-loop operation) or control values (open-loop operation) may exceed the respectively valid limit values:

- Open-loop operation: The limitation of the control value results from the parameters 0x07000005 and 0x07000006.
- Closed-loop operation: The currently valid limits can be queried with `CMN?` (p. 159) and `CMX?` (p. 161).

The amplitude is only limited during wave generator output: The corresponding limit value is output for points with a value that exceeds the respectively valid limit. An error code is **not** set.

INFORMATION

The length of the waveform influences the frequency of the wave generator output.

➢ Define the waveform so that the following conditions are met:
  - The frequency of the wave generator output is lower than the maximum permissible operating frequency of the connected mechanics (refer to the specifications for the mechanics).
  - The frequency of the wave generator output is selected so that the motor driver in the C-413 does not overheat (when overheating occurs, the output current is automatically switched off).

Optional: Deleting the wave table content

1. Make sure that the selected wave table is **not** connected to a wave generator for which the output has been started. Refer to "Configuring a Wave Generator" (p. 132) and "Stopping the Wave Generator Output" (p. 137) for details.
2. Delete the content of the wave tables with the `WCL` command (p. 223).

   The complete content of the selected wave table is deleted. It is **not** possible to delete the wave table content one segment at a time.

INFORMATION

When the C-413 is switched off or rebooted, the wave table content is automatically deleted.

Examples for creating waveforms

The following examples will help you to create the waveform.

INFORMATION

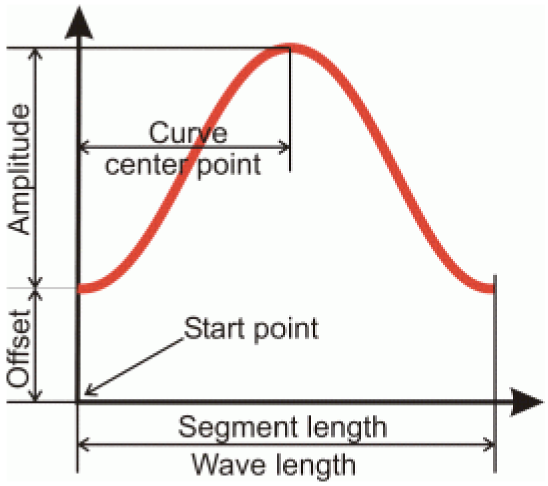
The offset defined with the `WAV` command only refers to one waveform (one segment). The offset defined with the `WOS` command (p. 227) is added to all waveforms that are output by the wave generator.

Sine curve 1

- Symmetrical sine curve with offset
- Segment overwrites the wave table content

Command: `WAV 2 X SIN_P 2000 20 10 2000 0 1000`

<WaveTableID> = 2
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 2000
<Amp> = 20
<Offset> = 10
<WaveLength> = 2000
<StartPoint> = 0
<CurveCenterPoint> = 1000

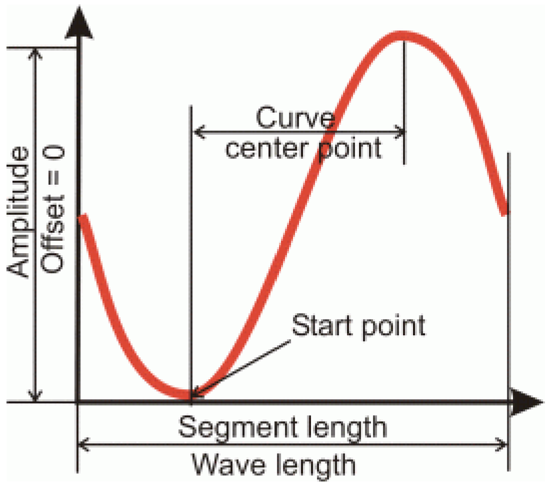


Sine curve 2

- Symmetrical sine curve without offset
- Segment overwrites the wave table content

Command: `WAV 2 X SIN_P 2000 30 0 2000 499 1000`

<WaveTableID> = 2
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 2000
<Amp> = 30
<Offset> = 0
<WaveLength> = 2000
<StartPoint> = 499
<CurveCenterPoint> = 1000

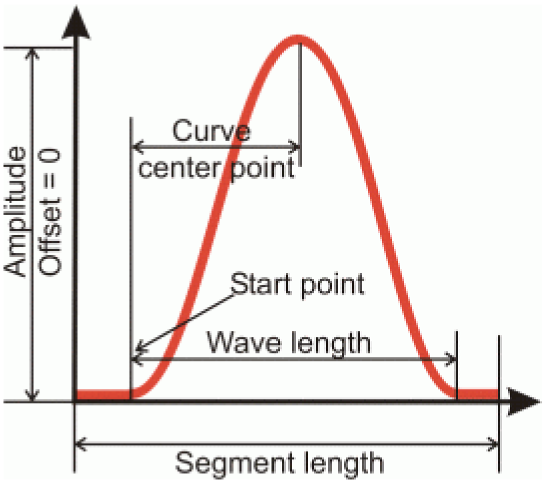


Sine curve 3

- Symmetrical sine curve without offset
- Segment is attached to the wave table content

Command: `WAV 2 & SIN_P 2000 25 0 1800 100 900`

<WaveTableID> = 2
<AppendWave> = &
<WaveType> = SIN_P
<SegLength> = 2000
<Amp> = 25
<Offset> = 0
<WaveLength> = 1800
<StartPoint> = 100
<CurveCenterPoint> = 900

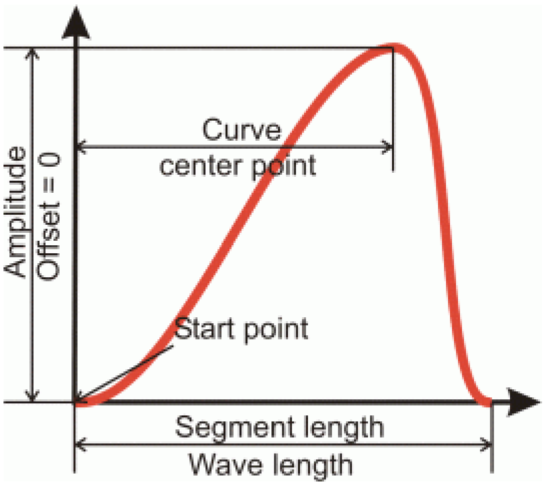


Sine curve 4

- Asymmetrical curve without offset
- Segment overwrites the wave table content

Command: `WAV 3 X SIN_P 4000 20 0 4000 0 3100`

<WaveTableID> = 3
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 4000
<Amp> = 20
<Offset> = 0
<WaveLength> = 4000
<StartPoint> = 0
<CurveCenterPoint> = 3100

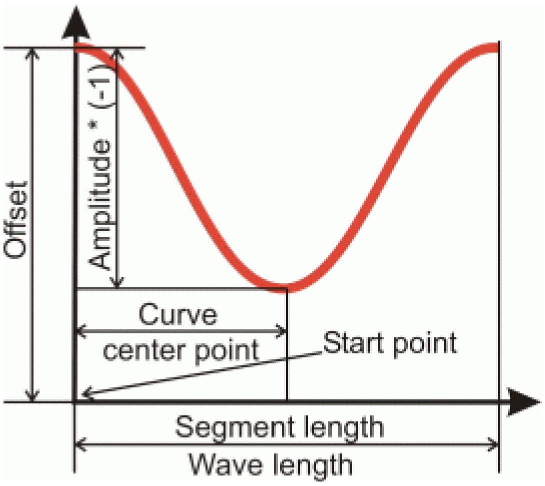


Sine curve 5

- Symmetrical curve with negative amplitude
- Segment overwrites the wave table content

Command: `WAV 1 X SIN_P 1000 -30 45 1000 0 500`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 1000
<Amp> = –30
<Offset> = 45
<WaveLength> = 1000
<StartPoint> = 0
<CurveCenterPoint> = 500

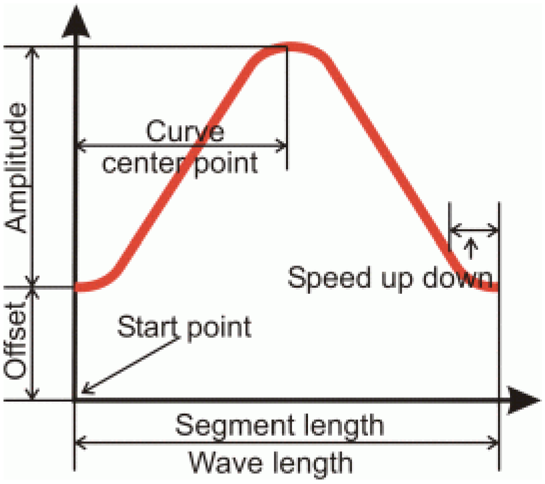


Ramp curve 1

- Symmetrical ramp curve with offset
- Segment overwrites the wave table content

Command: `WAV 4 X RAMP 2000 20 10 2000 0 100 1000`

<WaveTableID> = 4
<AppendWave> = X
<WaveType> = RAMP
<SegLength> = 2000
<Amp> = 20
<Offset> = 10
<WaveLength> = 2000
<StartPoint> = 0
<SpeedUpDown> = 100
<CurveCenterPoint> = 1000

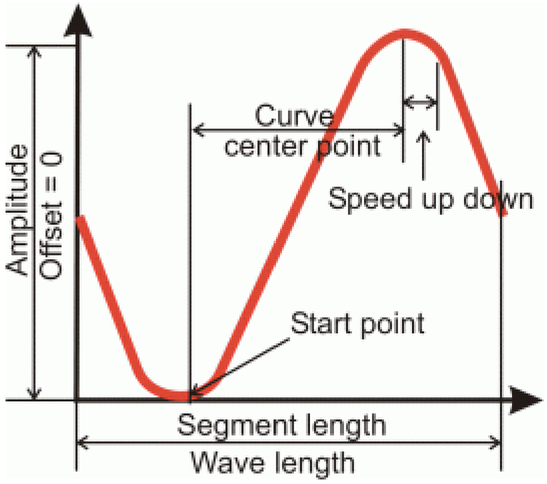


Ramp curve 2

- Symmetrical ramp curve without offset
- Segment overwrites the wave table content

Command: `WAV 4 X RAMP 2000 35 0 2000 499 100 1000`

<WaveTableID> = 4
<AppendWave> = X
<WaveType> = RAMP
<SegLength> = 2000
<Amp> = 35
<Offset> = 0
<WaveLength> = 2000
<StartPoint> = 499
<SpeedUpDown> = 100
<CurveCenterPoint> = 1000

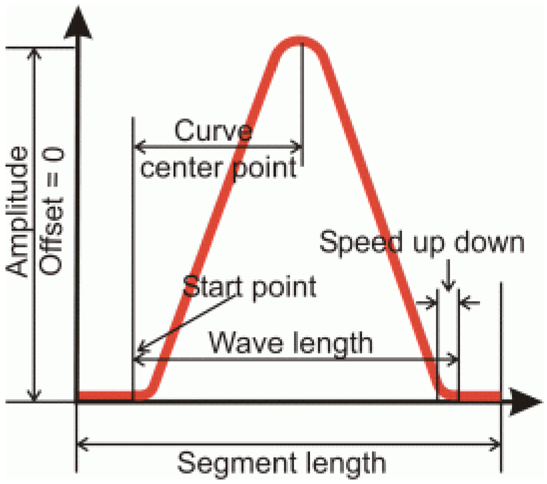


Ramp curve 3

- Symmetrical ramp curve without offset
- Segment overwrites the wave table content

Command: `WAV 5 X RAMP 2000 15 0 1800 120 50 900`

<WaveTableID> = 5
<AppendWave> = X
<WaveType> = RAMP
<SegLength> = 2000
<Amp> = 15
<Offset> = 0
<WaveLength> = 1800
<StartPoint> = 120
<SpeedUpDown> = 50
<CurveCenterPoint> = 900

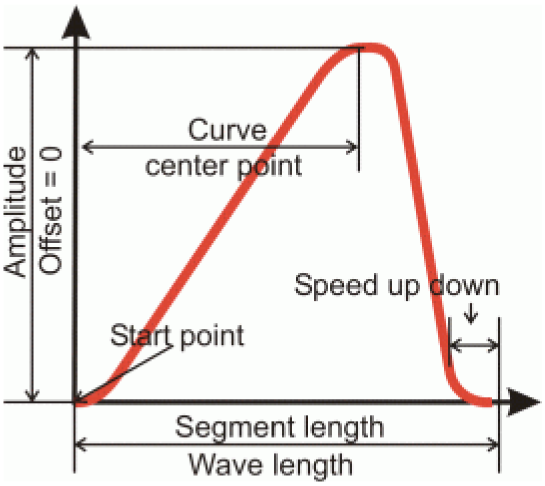


Ramp curve 4

- Asymmetrical ramp curve without offset
- Segment is attached to the wave table content

Command: `WAV 5 & RAMP 3000 35 0 3000 0 200 2250`

<WaveTableID> = 5
<AppendWave> = &
<WaveType> = RAMP
<SegLength> = 3000
<Amp> = 35
<Offset> = 0
<WaveLength> = 3000
<StartPoint> = 0
<SpeedUpDown> = 200
<CurveCenterPoint> = 2250

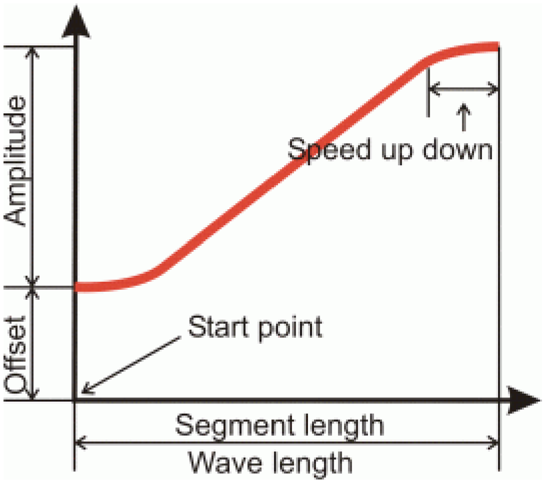


Single scan line 1

- Scan line with offset
- Segment overwrites the wave table content

Command: `WAV 1 X LIN 1500 30 15 1500 0 370`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = LIN
<SegLength> = 1500
<Amp> = 30
<Offset> = 15
<WaveLength> = 1500
<StartPoint> = 0
<SpeedUpDown> = 370

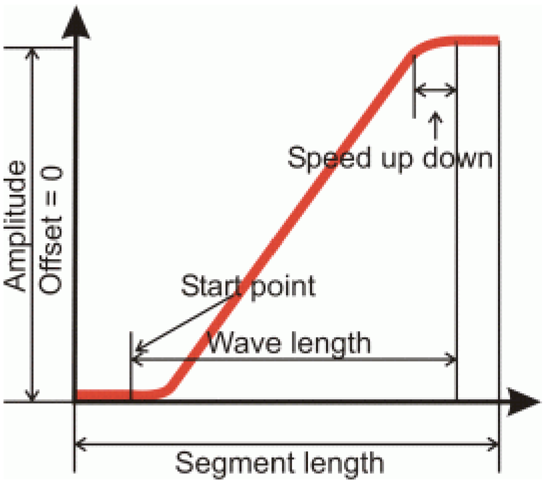


Single scan line 2

- Scan line without offset
- Segment overwrites the wave table content

Command: `WAV 2 X LIN 1500 40 0 1100 210 180`

<WaveTableID> = 2
<AppendWave> = X
<WaveType> = LIN
<SegLength> = 1500
<Amp> = 40
<Offset> = 0
<WaveLength> = 1100
<StartPoint> = 210
<SpeedUpDown> = 180

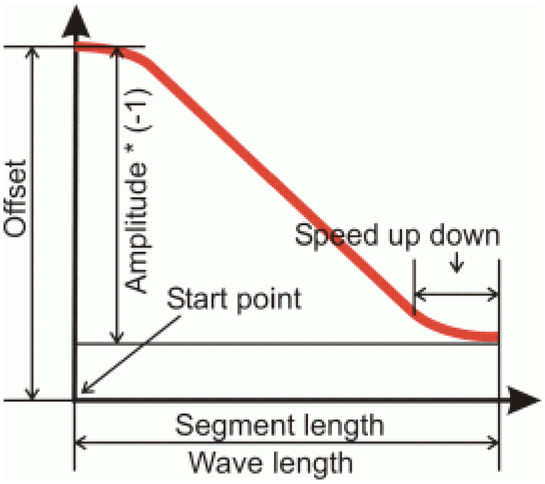


Single scan line 3

- Scan line with negative amplitude
- Segment is attached to the wave table content

Command: `WAV 2 & LIN 3000 -40 50 3000 0 650`

<WaveTableID> = 2
<AppendWave> = &
<WaveType> = LIN
<SegLength> = 3000
<Amp> = –40
<Offset> = 50
<WaveLength> = 3000
<StartPoint> = 0
<SpeedUpDown> = 650

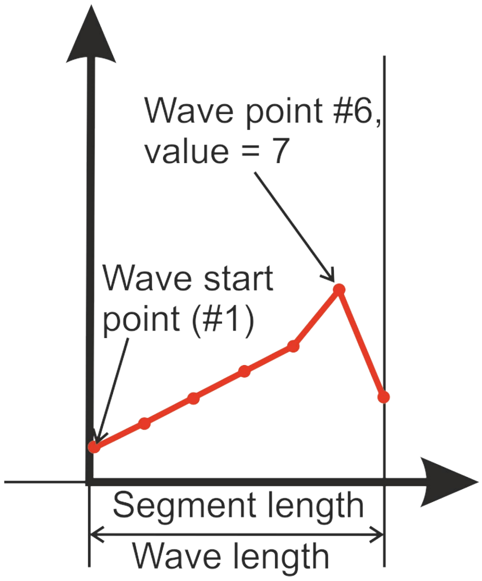


User-defined form

- User-defined curve
- Segment overwrites the wave table content

Command: `WAV 2 X PNT 1 7 1 2 3 4 5 7 3`

<WaveTableID> = 2
<AppendWave> = X
<WaveType> = PNT
<WaveStartPoint> = 1
<WaveLength> = 7
<WavePoint> = 1, 2, 3, 4, 5, 7, 3



7.6.4 Configuring a Wave Generator

The wave generator is configured with the following steps:

- Connecting or disconnecting a wave generator and a wave table (p. 133)
- Optional: Setting the output offset (p. 133)
- Optional: Setting the number of output cycles (p. 134)
- Optional: Setting the output rate (p. 134)

This manual contains an example for setting the output rate (p. 135).

INFORMATION

The following settings can only be changed in the volatile memory of the C-413 and are lost when the C-413 is switched off or rebooted:

- Assignment of wave tables to wave generators
- Number of output cycles of the wave generator

The following settings can be stored in the nonvolatile memory of the C-413 with the `WPA` command (p. 229):

- Output offset for the wave generator
- Wave generator table rate

Connecting or disconnecting a wave generator and a wave table

- Get the current connection of the wave generator and wave table with the `WSL?` command (p. 231).
- Connect or disconnect the wave generator and the wave table:

  a) Make sure that the output has **not** been started for the selected wave generator. Refer to "Stopping the wave generator output" (p. 137) for details.

  b) Use the `WSL` command (p. 230) to connect the selected wave table with the selected wave generator or terminate the connection of the selected generator to a wave table.

  Two or more generators can be connected to the same wave table, but a generator cannot be connected to more than one wave table.

Optional: Setting the output offset

The output offset is specified by the ***Wave Offset*** parameter (ID 0x1300010B) and added to the current wave value during the wave generator output:

Generator output = output offset + current wave value

Do not confuse the output offset value with the offset settings specified with the `WAV` command (p. 217) when the waveform is created. While the WAV offset affects only one segment (i.e. only one waveform), the output offset is added to all waveforms that are output by the wave generator.

- Determine the current output offset of the wave generator by getting the value of the ***Wave Offset*** parameter in the volatile memory:

  – Send the `WOS?` command (p. 228).

  \- or -

  – Get the value of the parameter with the `SPA?` command (p. 200).

- Determine the default setting for the output offset of the wave generator by getting the value of the ***Wave Offset*** parameter in the nonvolatile memory with the `SEP?` command (p. 198).
- Set the output offset by changing the value of the ***Wave Offset*** parameter:

  – Write the new output offset to the volatile memory with the `WOS` command (p. 227).

  \- or -

  a) Send the `CCL 1 advanced` command to go to command level 1.

  b) Change the output offset in the volatile memory with the `SPA` command (p. 199) or in the nonvolatile memory with the `SEP` command (p. 197).

  If you want to write the value of the parameter from the volatile to the nonvolatile memory of the C-413:

  c) Send the `CCL 1 advanced` command to go to command level 1.

  d) Save the parameter value with the `WPA` command (p. 229).

INFORMATION

In closed-loop operation, the interpretation of the output offset depends on the selected control mode. In open-loop operation, the output offset corresponds to the force to be applied in N (Refer to "Generating Control Values" (p. 24) and "Control Modes and Control Variables" (p. 28) for further information).

When the output offset is set, the resulting target values (closed-loop operation) or control values (open-loop operation) may exceed the respectively valid limit values:

- Open-loop operation: The limitation of the control value results from the parameters 0x07000005 and 0x07000006.
- Closed-loop operation: The currently valid limits can be queried with `CMN?` (p. 159) and `CMX?` (p. 161).

The amplitude is only limited during wave generator output: The corresponding limit value is output for points with a value that exceeds the respectively valid limit. An error code is **not** set.

Optional: Setting the number of output cycles

The factory default setting for the number of output cycles is 0. The waveform is output in the factory setting without a time limitation until it is stopped with the `WGO` (p. 224) or `#24` (p. 150) or `STP` (p. 204) command.

- ➢ Send the `WGC?` command (p. 224) to query the current setting for the number of output cycles of the wave generator.
- ➢ Set the number of output cycles of the wave generator with the `WGC` command (p. 223).

INFORMATION

When the number of output cycles is set during the wave generator output, the counting of the output cycles starts when the `WGC` command is sent.

Optional: Setting the output rate

The wave generator table rate is specified by the ***Wave Generator Table Rate*** parameter (ID 0x13000109). The individual output cycles of the waveform can be lengthened with the value of the parameter. The duration of an output cycle for the waveform can be calculated as follows:

Output duration = servo cycle time * output rate * number of points

where

- the servo cycle time for the C-413 is specified by the parameter 0x0E000200 (in seconds)
- the output rate is the number of servo cycles that the output of a waveform point lasts; the default is 1
- the number of points is the length of the waveform (i.e., the wave table length)

Different output rates can be set for the individual wave generators of the C-413.

- ➢ Determine the current wave generator table rate by getting the value of the ***Wave Generator Table Rate*** parameter in the volatile memory:

– Send the `WTR?` command (p. 232).

- or -

– Get the value of the parameter with the `SPA?` command (p. 200).

➢ Determine the default setting for the wave generator table rate by getting the value of the ***Wave Generator Table Rate*** parameter in the nonvolatile memory with the `SEP?` command (p. 198).

➢ Set the output rate by changing the value of the ***Wave Generator Table Rate*** parameter:

– Write the new output rate to the volatile memory with the `WTR` command (p. 231).

- or -

a) Send the `CCL 1 advanced` command to go to command level 1.

b) Change the output rate in the volatile memory with the `SPA` command (p. 199) or in the nonvolatile memory with the `SEP` command (p. 197).

If you want to write the value of the parameter from the volatile to the nonvolatile memory of the C-413:

c) Send the `CCL 1 advanced` command to go to command level 1.

d) Save the parameter value with the `WPA` command (p. 229).

INFORMATION

If the wave generator identifier 0 is used when setting the output rate with the `WTR` command, the output rate for all wave generators is set to the same value.

Example for setting the output rate

Action	Command	Result
Define a sine curve for wave table 2.	`WAV 2 X SIN_P 2000 20 10 2000 0 1000`	The length of the waveform and thus the number of points in the wave table is 2000.
Read the servo cycle time of the C-413.	`SPA? 1 0x0E000200`	The default setting for the servo cycle time of the C-413 is 200 µs
Read current output rate.	`WTR?`	Default value for the output rate = 1 (each point in the wave table is output during one servo cycle) Duration of an output cycle (see calculation formula above): 0.0002 s • 1 • 2000 = 0.4 s
Triple the number of servo cycles per point in the wave table for all wave generators.	`WTR 0 3 0`	Duration of an output cycle (see calculation formula above): 0.0002 s • 3 • 2000 = 1.2 s The C-413 does not support any interpolation. The last argument of the command, <InterpolationType>,

Action	Command	Result
		therefore has to be zero.

7.6.5 Starting and Stopping Output

Depending on the servo mode (p. 22), the wave generator outputs absolute target values or control values.

The wave generator output can be started immediately or by an external trigger signal:

- Starting the wave generator output immediately (p. 137)
- Starting the wave generator output with an external trigger signal (p. 137)

Further steps:

- Stopping the wave generator output (p. 137)
- Optional: Getting the activation state of the wave generator (p. 137)
- Optional: Using the start option "Start at the endpoint of the last cycle" (p. 138)
- Optional: Starting data recording during the wave generator output (p. 138)

This manual contains examples for starting/stopping the wave generator output (p. 139).

INFORMATION

When the wave generator is running for the corresponding axis, it is not permitted to switch servo mode (p. 22) on or off or to change the control mode (p. 28).

INFORMATION

The trigger configuration for the wave generator output is only present in the volatile memory of the C-413 and is lost when the C-413 is switched off or rebooted.

INFORMATION

Wave generator output and analog control input:
It is possible to configure an axis for control by an analog input line while the wave generator output is active for that axis. In that case, the wave generator will continue to be active but its output will no longer be used for generating target or control values. As long as the corresponding axis is set up to be commanded by analog control input, the wave generator output can be stopped but not restarted.

Wave generator output and motion commands:
When the wave generator output is active, motion commands such as CTV (p. 169), MOV (p. 188) or SVA (p. 205) are not allowed for the associated axis.

Refer to "Generating Control Values" (p. 24) for further information.

Requirements

✓ You have created the desired waveform (p. 124).

- ✓ You have connected the wave generator to the corresponding wave table (p. 133).
- ✓ If you want to use trigger signals to start the wave generator output and interrupt it if necessary:

  You have connected suitable digital input signals to the pins 9, 10, 11 or 12 of the **I/O** panel plug of the C-413 (p. 295).

Starting the wave generator output immediately

- ➢ Start the wave generator output with the `WGO` command (p. 224) by setting bit 0 ("Immediate start of the wave generator output" start mode).

  The output takes place synchronously with the servo cycles of the C-413.

  When the wave generator output is started, a data recording cycle automatically starts.

Starting the wave generator output with an external trigger signal

- ➢ Follow the instructions in ""Wave Generator" Trigger Mode – Starting the Wave Generator Output" (p. 93).

Stopping the wave generator output

- ➢ Stop the wave generator output by sending one of the following commands:
  - `WGO F 0`, where *F* specifies the wave generator and *0* causes stopping (p. 224).
  - `STP` (p. 204)
  - `#24` (p. 150)

  When the wave generator output is stopped by sending `STP` or `#24`, the C-413 sets the error code 10 (get with the `ERR?` command (p. 178)).

When the number of output cycles has been limited (p. 134), the wave generator output is automatically stopped when the specified number of cycles is reached.

INFORMATION

When the wave generator output has been started by an external trigger signal, it is interrupted or continued according to the state of the digital input line (p. 93). The interruption of the wave generator output by the digital input line is **not** the same as stopping.

INFORMATION

Exiting the PC software does **not** stop the wave generator output.

Optional: Getting the activation state of the wave generator

- ➢ Query whether the wave generator output is running with the `#9` command (p. 149).
- ➢ Query the last-commanded start settings (start mode and options) of the wave generator with the `WGO?` command (p. 226).

  Stopping the wave generator output with `#24` (p. 150) or `STP` (p. 204) sets the start mode value to zero.

Optional: Using the start option "Start at the endpoint of the last cycle"

- Start the wave generator output with the WGO command (p. 224) by setting bit 8 (start option "Start at the endpoint of the last cycle") in addition to bit 0 or 1 (start mode).

  When bit 8 is set, the second and all following output cycles each start at the endpoint of the previous cycle, which makes this start option suitable for scanning applications. For examples, see the following figures.

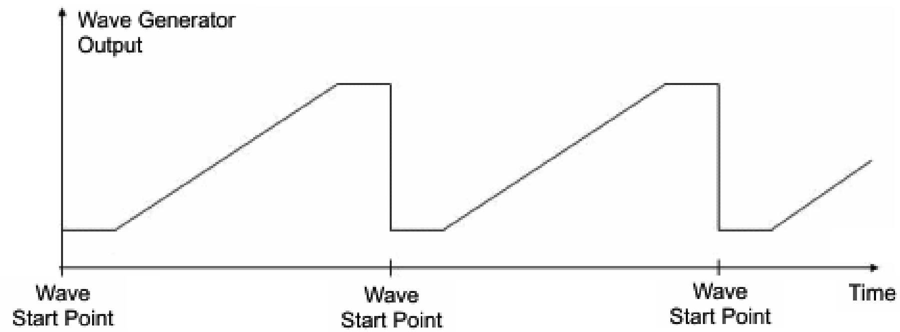


Figure 34: Wave generator output without the start option "Start at the endpoint of the last cycle"

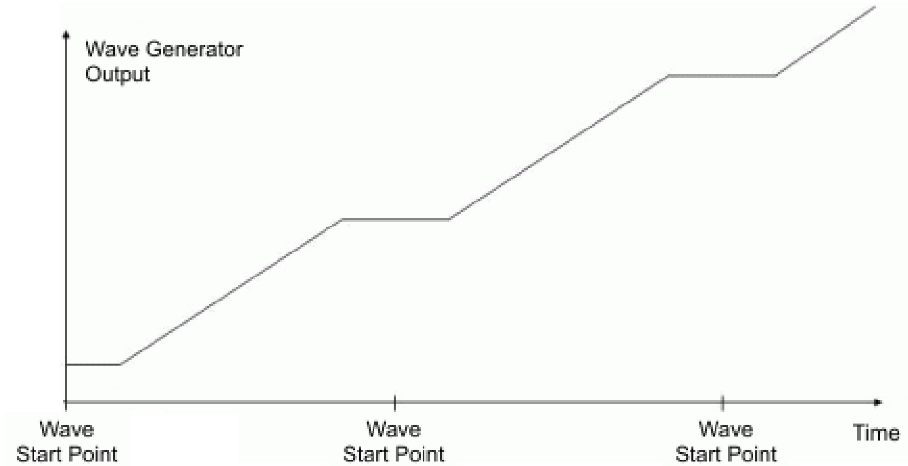


Figure 35: Wave generator output with the start option "Start at the endpoint of the last cycle"

Optional: Starting data recording during the wave generator output

- Start the data recording during the wave generator output by sending the WGR command (p. 227).

When the wave generator is started with the start mode "Immediate start of the wave generator output" (p. 137), the first data recording cycle will automatically start.

The recorded data can be read out with the DRR? command (p. 175). Refer to "Data Recorder" (p. 83) for more information.

Example for starting/stopping the wave generator output

Action	Command	Result
Define a sine curve for wave table 4.	`WAV 4 X SIN_P 2000 2 0 2000 0 1000`	The length of the waveform and therefore the number of points in the wave table is 2000.
Connect wave generator 1 to wave table 4.	`WSL 1 4`	Requirement for wave generator output fulfilled: No wave generator output is possible without allocation of a wave table.
Start wave generator 1 immediately.	`WGO 1 1`	The waveform defined in wave table 4 is output.
Stop wave generator 1.	`WGO 1 0`	The output of the waveform points is stopped.

Refer to ""Wave Generator" Trigger Mode – Starting the Wave Generator Output" (p. 93) for a further example.

7.7 Optimizing the Servo Cycle Time

NOTICE

Servo cycle time too low!

If the servo cycle time of the C-413 is set to a value that is too low, the utilization of the processor of the C-413 can be too high. Excessive load on the processor can lead to loss of communication with the C-413.

- ➢ Only reduce the servo cycle time of the C-413 when necessary.
- ➢ Check the utilization of the processor by measuring the signal at digital output 6 (pin 20 of the **I/O** panel plug (p. 295)). The utilization is too high when the signal is at the high level more than 70% of the time.

When the utilization is too high:

- ➢ Check the communication with the C-413.

When the communication is faulty or has been terminated:

1. Reboot the C-413.
2. Make sure that the C-413 is **not** performing any time-intensive tasks.
3. Extend the servo cycle time; refer to "Optimizing the Servo Cycle Time of the C-413".

INFORMATION

The servo cycle time of the C-413 influences, among other things, the possible sampling rate, the data recording, (p. 83) and the output of the wave generators (p. 120).

The servo cycle time is specified by the value of the ***Servo Update Time*** parameter (ID 0x0E000200) (in seconds).

- Maximum servo cycle time: 200 µs (default setting)
- Minimum servo cycle time: 100 µs

The servo cycle time can be changed within the permissible range in steps of 25 µs. To change the servo cycle time, it is necessary to change the ***Servo Update Time*** parameter in the nonvolatile memory and then reboot the C-413; refer to "Optimizing the servo cycle time of the C-413".

INFORMATION

You can make the settings for optimizing the servo cycle time with commands. Alternatively, you have access to the parameters that are mentioned in the following via the *System Global* parameter group in the ***Device Parameter Configuration*** window of PIMikroMove.

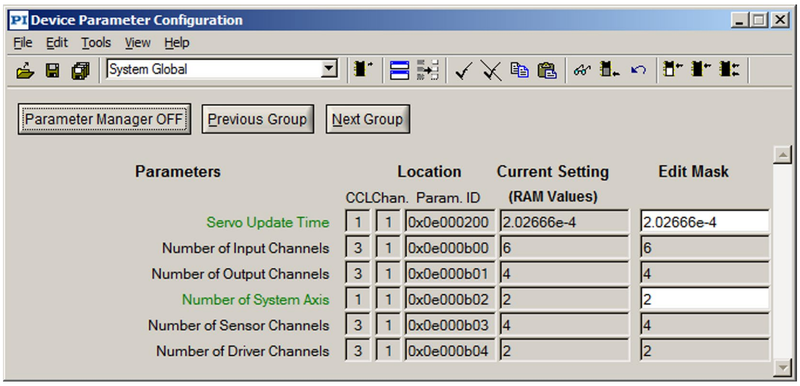


Figure 36: Example: The Device Parameter Configuration window in PIMikroMove; the System Global parameter group is shown

Requirements

✓ You have connected a measuring device to digital output 6 of the **I/O** panel plug (p. 66).

Tools and accessories

- Suitable measuring device, e.g., digital storage oscilloscope

Optimizing the servo cycle time of the C-413

When you make the setting in the ***Device Parameter Configuration*** window of PIMikroMove:

➢ Read "Device Parameter Configuration" in the PIMikroMove manual.

➢ Pay attention to the following instructions when working with commands, but do not determine or modify the parameter value with the specified commands but instead, in the ***Device Parameter Configuration*** window.

➢ When prompted, enter the password *advanced* to go to command level 1.

If you make the setting by entering commands (e.g., in PITerminal or in the ***Command entry*** window of PIMikroMove):

1. Make sure that the C-413 is **not** performing any time-intensive tasks.

In addition to moving axes, time-consuming tasks also include e. g. reading recorded data and the wave generator output.

2. Check the processor workload by measuring the signal at digital output 6.

   The signal is active high. Each servo cycle starts with the positive edge of the signal and ends with the negative edge. The portion of time where the signal has the high level specifies the workload of the processor:

   – High level time ≤ 70%: Optimal workload
   – High level time > 70%: Excessive workload

   Is the workload too high?

   – If yes: Optimize the servo cycle time; see step 4.
   – If no: Repeat the check with a higher workload; see step 3.

3. Check processor workload with a higher load:

   a) Start a time-intensive task, e.g., the wave generator output (p. 120) for both axes.
   b) Measure the signal at digital output 6.

   Is the utilization optimum at a higher load?

   – If yes: You can leave the servo cycle time unchanged or reduce it by 25 µs. If you want to leave the servo cycle time unchanged, the optimization procedure is finished. To reduce the servo cycle time, see step 4.
   – If no: Increase the servo cycle time, see step 4.

4. Optimize the servo cycle time:

   a) Send the `CCL 1 advanced` command to go to command level 1.
   b) Get the value of the servo cycle time in the nonvolatile memory by sending the command `SEP? 1 0x0E000200`.
   c) Change the value of the servo cycle time in the nonvolatile memory: Send the command `SEP 100 1 0x0E000200 Time`, where *Time* indicates the new servo cycle time in seconds and, depending on the result of the utilization test, is 25 µs lower or higher than the value queried in step b.
   d) Reboot the C-413 by sending the `RBT?` command.
   e) Re-establish the communication between the C-413 and the PC.
   f) Repeat steps 1 to 3 in this order.

8 GCS Commands

8.1 Notation

The following notation is used to define the GCS syntax and to describe the commands:

<...>	Angle brackets indicate an argument of a command, can be an element identifier or a command-specific parameter.
[...]	Square brackets indicate an optional entry.
{...}	Curly brackets indicate a repetition of entries, i.e., it is possible to access more than one element (e.g., several axes) in one command line.
LF	LineFeed (line feed, ASCII character 10), is the default termination character (character at the end of a command line).
SP	Space (ASCII character 32) indicates a space.
"..."	Quotation marks indicate that the characters enclosed are returned or to be entered.

8.2 GCS Syntax for Syntax Version 2.0

A GCS command consists of 3 characters, e.g., CMD. The corresponding query command has a question mark at the end, e.g., CMD?

Command mnemonic:

CMD ::= character1 character2 character3 [?]

Exceptions:

- Single-character commands, e.g., fast query commands, consist only of one ASCII character. The ASCII character is written as combination of # and the character code in decimal format, e.g., as `#24`.
- `*IDN?` (for GPIB compatibility).

The command mnemonic is not case-sensitive. The command mnemonic and all arguments (e.g., axis identifiers, channel identifiers, parameters, etc.) must be separated from each other by a space (SP). The command line has to be terminated with a line feed (LF).

CMD[{{SP}<Argument>}]LF

CMD?[{{SP}<Argument>}]LF

Exception:

- Single-character commands are not followed by a termination character. However, the response to a single-character command is followed by a termination character.

The argument <AxisID> is used for the logical axes of the controller. Depending on the controller, an axis identifier can consist of up to 16 characters. All alphanumeric characters and the underscore are allowed. Refer to "Commandable Elements" (p. 13) for the identifiers supported by the C-413.

Example 1:

Axis 1 is to be moved to position 10.0. The unit depends on the controller (e.g., µm or mm).

Send: `MOV`SP`1`SP`10.0`LF

More than one command mnemonic per line is not allowed. Several groups of arguments following a command mnemonic are allowed.

Example 2:

Two axes connected to the same controller are to be moved:

Send: `MOV`SP`1`SP`17.3`SP`2`SP`2.05`LF

When a part of a command line cannot be executed, the line is not executed at all.

When all arguments are optional and are not specified, the command is executed for all possible argument values.

Example 3:

All parameters in the volatile memory are to be reset.

Send: `RPA`LF

Example 4:

The position of all axes is to be queried.

Send: `POS?`LF

The response syntax is as follows:

[<Argument>[{SP<Argument>}]"="]<Value>LF

With multi-line replies, the space preceding the termination character is left out of the last line:

{[<Argument>[{SP<Argument>}]"="]<Value>SPLF}

[<Argument>[{SP<Argument>}]"="]<Value>LF for the last line!

The arguments are listed in the response in the same order as in the query command.

Query command:

CMD?SP<Arg3>SP<Arg1>SP<Arg2>LF

Response to this command:

<Arg3>"="<Val3>SPLF

<Arg1>"="<Val1>SPLF

<Arg2>"="<Val2>LF

Example 5:

Send: `TSP?`SP`2`SP`1`LF

Receive: `2=-1158.4405`SPLF

`1=+0000.0000`LF

INFORMATION

With the C-413, each command line can contain a command mnemonic and up to 12 arguments.

8.3 Command Overview

Command	Format	Description
#5 (p. 149)	#5	Request Motion Status
#7 (p. 149)	#7	Request Controller Ready Status
#9 (p. 149)	#9	Get Wave Generator Status
#24 (p. 150)	#24	Stop All Axes
*IDN? (p. 151)	*IDN?	Get Device Identification
AOS (p. 151)	AOS {<AxisID> <Offset>}	Set Analog Input Offset
AOS? (p. 153)	AOS? [{<AxisID>}]	Get Analog Input Offset
ATZ (p. 154)	ATZ [{<AxisID> <LowValue>}]	Set Automatic Zero Point Calibration
ATZ? (p. 156)	ATZ? [{<AxisID>}]	Get State Of Automatic Zero Point Calibration
CAV? (p. 156)	CAV? [{<AxisID>}]	Get Current Value Of Controlled Variable
CCL (p. 157)	CCL <Level> [<PSWD>]	Set Command Level
CCL? (p. 158)	CCL?	Get Command Level
CCV? (p. 158)	CCV? [{<AxisID>}]	Get Control Value
CMN? (p. 159)	CMN? [{<AxisID>}]	Get Minimum Commandable Closed-Loop Target
CMO (p. 159)	CMO {<AxisID> <CtrlMode>}	Set Closed-Loop Control Mode
CMO? (p. 161)	CMO? [{<AxisID>}]	Get Closed-Loop Control Mode

Command	Format	Description
CMX? (p. 161)	CMX? [{<AxisID>}]	Get Maximum Commandable Closed-Loop Target
CST? (p. 162)	CST? [{<AxisID>}]	Get Assignment Of Stages To Axes
CSV? (p. 163)	CSV?	Get Current Syntax Version
CTI (p. 163)	CTI {<TrigInID> <CTIPam> <Value>}	Set Configuration Of Trigger Input
CTI? (p. 165)	CTI? [{<TrigInID> <CTIPam>}]	Get Configuration Of Trigger Input
CTO (p. 165)	CTO {<TrigOutID> <CTOPam> <Value>}	Set Configuration Of Trigger Output
CTO? (p. 167)	CTO? [{<TrigOutID> <CTOPam>}]	Get Configuration Of Trigger Output
CTR (p. 168)	CTR {<AxisID> <TargetRelative>}	Set Target Relative To Current Closed-Loop Target
CTV (p. 169)	CTV {<AxisID> <TargetAbsolute>}	Set Absolute Closed-Loop Target
CTV? (p. 170)	CTV? [{<AxisID>}]	Get Closed-Loop Target
DIO (p. 171)	DIO {<DIOID> <OutputOn>}	Set Digital Output Lines
DIO? (p. 171)	DIO? [{<DIOID>}]	Get Digital Input Lines
DRC (p. 172)	DRC {<RecTableID> <Source> <RecOption>}	Set Data Recorder Configuration
DRC? (p. 174)	DRC? [{<RecTableID>}]	Get Data Recorder Configuration
DRL? (p. 174)	DRL? [{<RecTableID>}]	Get Number Of Recorded Points
DRR? (p. 175)	DRR? [<StartPoint> <NumberOfPoints> [{<RecTableID>}]]	Get Recorded Data Values
DRT (p. 176)	DRT {<RecTableID> <TriggerSource> <Value>}	Set Data Recorder Trigger Source
DRT? (p. 177)	DRT? [{<RecTableID>}]	Get Data Recorder Trigger Source
ERR? (p. 178)	ERR?	Get Error Number
FRF (p. 178)	FRF [{<AxisID>}]	Fast Reference Move To Reference Switch
FRF? (p. 180)	FRF? [{<AxisID>}]	Get Referencing Result
GWD? (p. 180)	GWD? [<StartPoint> <NumberOfPoints> [{<WaveTableID>}]]	Get Wave Table Data
HDR? (p. 181)	HDR?	Get All Data Recorder Options
HLP? (p. 183)	HLP?	Get List of Available Commands
HLT (p. 183)	HLT [{<AxisID>}]	Halt Motion Smoothly
HPA? (p. 184)	HPA?	Get List Of Available Parameters
HPV? (p. 185)	HPV?	Get Parameter Value Description
IDN? (p. 186)	IDN?	Get Device Identification
IMP (p. 187)	IMP <AxisID> <Amplitude>	Start Impulse and Response Measurement
LIM? (p. 188)	LIM? [{<AxisID>}]	Indicate Limit Switches
MOV (p. 188)	MOV {<AxisID> <Position>}	Set Target Position

Command	Format	Description
MOV? (p. 189)	MOV? [{<AxisID>}]	Get Target Position
MVR (p. 189)	MVR {<AxisID> <Distance>}	Set Target Relative To Current Position
ONT? (p. 191)	ONT? [{<AxisID>}]	Get On-Target State
OVF? (p. 191)	OVF? [{<AxisID>}]	Get Overflow State
POS (p. 192)	POS {<AxisID> <Position>}	Set Real Position
POS? (p. 193)	POS? [{<AxisID>}]	Get Real Position
PUN? (p. 193)	PUN? [{<AxisID>}]	Get Axis Unit
RBT (p. 194)	RBT	Reboot System
RON (p. 194)	RON {<AxisID> <ReferenceOn>}	Set Reference Mode
RON? (p. 194)	RON? [{<AxisID>}]	Get Reference Mode
RPA (p. 195)	RPA [{<ItemID> <PamID>}]	Reset Volatile Memory Parameters
RTR (p. 195)	RTR <RecordTableRate>	Set Record Table Rate
RTR? (p. 196)	RTR?	Get Record Table Rate
SAI? (p. 196)	SAI? [ALL]	Get List Of Current Axis Identifiers
SEP (p. 197)	SEP <Pswd> {<ItemID> <PamID> <PamValue>}	Set Non-Volatile Memory Parameters
SEP? (p. 198)	SEP? [{<ItemID> <PamID>}]	Get Non-Volatile Memory Parameters
SPA (p. 199)	SPA {<ItemID> <PamID> <PamValue>}	Set Volatile Memory Parameters
SPA? (p. 200)	SPA? [{<ItemID> <PamID>}]	Get Volatile Memory Parameters
SRG? (p. 201)	SRG? {<AxisID> <RegisterID>}	Query Status Register Value
STE (p. 202)	STE <AxisID> <Amplitude>	Start Step And Response Measurement
STP (p. 204)	STP	Stop All Axes
SVA (p. 205)	SVA {<AxisID> <ControlValueAbs>}	Set Absolute Open-Loop Control Value
SVA? (p. 206)	SVA? [{<AxisID>}]	Get Open-Loop Control Value
SVO (p. 206)	SVO {<AxisID> <ServoState>}	Set Servo Mode
SVO? (p. 207)	SVO? [{<AxisID>}]	Get Servo Mode
SVR (p. 208)	SVR {<AxisID> <ControlValueRel>}	Set Relative Open-Loop Control Value
TAD? (p. 208)	TAD? [{<InputSignalID>}]	Get ADC Value Of Input Signal
TIO? (p. 209)	TIO?	Tell Digital I/O Lines
TMN? (p. 210)	TMN? [{<AxisID>}]	Get Minimum Commandable Position
TMX? (p. 210)	TMX? [{<AxisID>}]	Get Maximum Commandable Position
TNR? (p. 210)	TNR?	Get Number Of Record Tables
TNS? (p. 211)	TNS? [{<InputSignalID>}]	Get Normalized Input Signal Value
TPC? (p. 211)	TPC?	Get Number of Output Signal Channels
TRI (p. 211)	TRI {<TrigInID> <TrigInMode>}	Set Trigger Input State
TRI? (p. 212)	TRI? [{<TrigInID>}]	Get Trigger Input State
TRO (p. 212)	TRO {<TrigOutID> <TrigMode>}	Set Trigger Output State

Command	Format	Description
TRO? (p. 213)	TRO? [{<TrigOutID>}]	Get Trigger Output State
TRS? (p. 213)	TRS? [{<AxisID>}]	Indicate Reference Switch
TSC? (p. 214)	TSC?	Get Number of Input Signal Channels
TSP? (p. 214)	TSP? [{<InputSignalID>}]	Get Input Signal Value
TWG? (p. 214)	TWG?	Get Number of Wave Generators
VEL (p. 215)	VEL {<AxisID> <Velocity>}	Set Closed-Loop Velocity
VEL? (p. 216)	VEL? [{<AxisID>}]	Get Closed-Loop Velocity
VOL? (p. 217)	VOL? [{<OutputSignalID>}]	Get Value Of Output Signal
WAV (p. 217)	WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>	Set Waveform Definition
WAV? (p. 223)	WAV? [{<WaveTableID> <WaveParameterID>}]	Get Waveform Definition
WCL (p. 223)	WCL {<WaveTableID>}	Clear Wave Table Data
WGC (p. 223)	WGC {<WaveGenID> <Cycles>}	Set Number Of Wave Generator Cycles
WGC? (p. 224)	WGC? [{<WaveGenID>}]	Get Number Of Wave Generator Cycles
WGO (p. 224)	WGO {<WaveGenID> <StartMode>}	Set Wave Generator Start/Stop Mode
WGO? (p. 226)	WGO? [{<WaveGenID>}]	Get Wave Generator Start/Stop Mode
WGR (p. 227)	WGR	Starts Recording In Sync With Wave Generator
WOS (p. 227)	WOS {<WaveGenID> <Offset>}	Set Wave Generator Output Offset
WOS? (p. 228)	WOS? [{<WaveGenID>}]	Get Wave Generator Output Offset
WPA (p. 229)	WPA <Pswd> [{<ItemID> <PamID>}]	Save Parameters To Non-Volatile Memory
WSL (p. 230)	WSL {<WaveGenID> <WaveTableID>}	Set Connection Of Wave Table To Wave Generator
WSL? (p. 231)	WSL? [{<WaveGenID>}]	Get Connection Of Wave Table To Wave Generator
WTR (p. 231)	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}	Set Wave Generator Table Rate
WTR? (p. 232)	WTR? [{<WaveGenID>}]	Get Wave Generator Table Rate

8.4 Command Descriptions for GCS 2.0

#5 (Request Motion Status)

Description: Queries the motion status of the axes.

Format: #5

Arguments: None

Response: The response <uint> is bit-mapped and returned as the hexadecimal sum of the following codes:

1=First axis in motion
2=Second axis in motion
4=Third axis in motion
...
0 indicates that all axes have finished moving.

#7 (Request Controller Ready Status)

Description: Queries the controller for ready state (tests if controller is ready to do a new command).

Note: Use #5 (p. 149) instead of #7 to verify if motion has finished.

Format: #7

Arguments: None

Response: B1h (ASCII character 177 = "±" in Windows) if controller is ready

B0h (ASCII character 176 = "°" in Windows) if controller is not ready
(e.g., executing a referencing move)

Troubleshooting: The response characters may be displayed differently in non-Western character sets or other operating systems.

#9 (Get Wave Generator Status)

Description: Requests the status of the wave generator(s).

Format: #9

Arguments: None

Response: The <uint> response is bit-mapped and output as the hexadecimal sum of the following codes:
1 = Wave generator 1 is active,
2 = Wave generator 2 is active,

	4 = Wave generator 3 is active, etc. "Active" = Wave generator output is running
Examples:	0 indicates that no wave generator is running 5 indicates that wave generators 1 and 3 are running

#24 (Stop All Axes)

Description: Stops all axes abruptly. See the notes below for further details.

Sets error code to 10.

This command is identical in function to STP (p. 204), but only one character is sent via the interface.

Format: #24

Arguments: None

Response: None

Notes: #24 and STP stop all axis motion caused by motion commands (e.g. CTV (p. 169), CTR (p. 168), MOV (p. 188), VEL (p. 215), SVA (p. 205), SVR (p. 208)), commands for referencing (FRF (p. 178)), the wave generator (WGO (p. 224)), an analog control input and the autozero procedure (ATZ (p. 154)).

The target value for the stopped axes in closed-loop operation is set as follows:

- The control variable is the position: The target value is set to the current value of the position.
- The control variable is the velocity or the force: The target value is set to zero.

The control values for the stopped axes in open-loop operation are each set to the value of the ***autozero Result*** parameter (ID 0x07000A03, see ATZ (p. 154)).

When the analog input is used as control source and the axis motion is stopped with STP (p. 204) or #24 (p. 150), the analog input channel is disconnected from the axis. To recommence commanding the axis via the analog input, the corresponding input signal channel must be reconnected to the axis. Refer to "Analog Input Signals" (p. 94) for further information.

*IDN? (Get Device Identification)

Description: Reports the device identity number.

Format: *IDN?

Arguments: None

Response: Single-line text terminated with a termination character (line feed) with controller name, serial number, and firmware version

Notes: For the C-413, *IDN? responds something like:

```
(c)2013-2014 Physik Instrumente (PI)
GmbH & Co. KG,C-413.2GA,0114009224,1.000
```

AOS (Set Analog Input Offset)

Description: Sets an offset to be added to the scaled value of the analog input for the specified axis.

Format: AOS {<AxisID> <Offset>}

Arguments: <AxisID> is one axis of the controller

<Offset> is the offset value, any floating point number. See below for details.

Response: None

Troubleshooting: Illegal axis identifier

Notes: AOS sets the value of the ***Analog Target Offset*** parameter (ID 0x06000501) in the volatile memory. Required command level: 0. (Setting the parameter value with SPA (p. 199) requires switching to command level 1 with CCL (p. 157).)

If the settings made with AOS are to be maintained when the C-413 is switched off or rebooted, they have to be saved with WPA (p. 229), refer also to "Adapting Settings" (p. 257).

In closed-loop operation, the interpretation of the offset depends on the selected control mode (p. 28). In open-loop operation, the offset corresponds to the force in N to be generated, refer also to "Output matrix" (p. 19).

The offset is only effective when an input signal channel of the controller is connected to the axis for triggering motions. The connection can be made via the ***ADC Channel for Target*** parameter (0x06000500) with SPA (p. 199) or SEP (p. 197).

When the offset is effective, it is included in the triggering of motions as follows:

- Closed-loop operation:
  Target value = scaled analog input value of the input signal channel + offset
  The target value can be queried with CTV? (p. 170).
- Open-loop operation:
  Control value = scaled analog input value of the input signal channel + offset
  The control value can be queried with CCV? (p. 158) and SVA? (p. 206).

When the resulting target value (closed-loop operation) or control value (open-loop operation) exceeds the respectively valid limit, the corresponding limit value is used. An error code is **not** set.

- Open-loop operation: The limitation of the control value results from the parameters 0x07000005 and 0x07000006.
- Closed-loop operation: The currently valid limits can be queried with CMN? (p. 159) and CMX? (p. 161).

Refer also to "Generating Control Values" (p. 24) and "Analog Input Signals" (p. 94).

Example: A C-413.2GA is used that can control 2 axes and is equipped with 2 analog input lines. In this example, axis 2 of the C-413 is in closed-loop operation.

Switch to command level 1 before you change parameter values with SPA or SEP.

Send: CCL 1 advanced

Select input signal channel 5 (**I/O** panel plug (p. 295)) as the control source for axis 2. The target value for axis 2 will now result from the scaled input value of channel 5 plus the offset.

Send: SPA 2 0x06000500 5

Set the digital offset of axis 2 to zero.

Send: AOS 2 0.0

Get the filtered and scaled value of input signal channel 5.

Send: TSP? 5

Receive: 5=3.22

Get the current target value of axis 2. The target value and the scaled value of input signal channel 5 are identical, since the offset is zero.

Send: CTV? 2
Receive: 2=3.22

Set offset of axis 2 to 1.5.
Send: AOS 2 1.50

The target value of axis 2 is the scaled value of input signal channel 5 plus the offset of axis 2.
Send: TSP? 5
Receive: 5=3.22
Send: CTV? 2
Receive: 2=4.72

As long as the target value of axis 2 is specified by an analog input, it is not possible to set the target value using commands e.g. CTV.
Send: CTV 2 6.0
Send: ERR?
Receive: 72

Disconnect the analog input from axis 2.
Send: SPA 2 0x06000500 0
The target value for axis 2 can now be set, e.g., with the CTV command. The AOS setting is no longer effective for generating the target value of axis 2.

AOS? (Get Analog Input Offset)

Description:	Queries the currently valid offset to the scaled value of the analog input for the specified axis (value of the ***Analog Target Offset*** parameter in the volatile memory (ID 0x06000501)).
Format:	AOS? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	{<AxisID>"="<Offset> LF} where <Offset> is the offset value; see AOS (p. 151) for further information
Troubleshooting:	Illegal axis identifier

ATZ (Set Automatic Zero Point Calibration)

Description:	Automatic zero point adjustment. Starts an adjustment procedure in which the axis is moved. The adjustment procedure can be stopped with #24 (p. 150) or STP (p. 204). The success of the automatic zero point adjustment can be queried with ATZ? (p. 156). The automatic zero point adjustment can take several seconds. Ask with #5 (p. 149) if the procedure is finished.
Format:	ATZ [{<AxisID> <LowValue>}]
Arguments	<AxisID> is one axis of the controller <LowValue> specifies at what position the control value is to be determined that is required to generate a force of 0 N in open-loop operation. This can also be NaN ("not a number"); in this case, the value of the A***utozero Low Value*** parameter (ID 0x07000A00) is used.
Response:	None
Troubleshooting:	Illegal axis identifier <LowValue> is not specified.
Notes:	The adjustment procedure started with ATZ defines the control value at which the axis generates a force of 0 N in open-loop operation. Adjustment is necessary, e.g., in the case of a vertically oriented motion axis, to compensate for the weight force of the moving part of the mechanics in open-loop operation. Note: Depending on the setting of the used parameters, the motion can extend over the entire travel range of the axis during the adjustment procedure. Make sure that the axis can move safely. For the axis to be adjusted, a successful referencing move must be executed before ATZ is used (start with FRF (p. 178)), or the current position must be set with POS (p. 192). When an input signal channel is active as the control source, it is automatically deactivated at the start of the autozero procedure and reactivated again after the procedure. The ***Power Up Autozero Enable*** parameter (ID 0x07000802) can be used to configure the C-413 so that

the autozero procedure is automatically performed after switching on or rebooting.

Procedure details:

The axis is moved to adjust the control value and force to each other. The motion range is defined by the <LowValue> value in the ATZ command (lower limit) and the value of the ***autozero High Value*** parameter (ID 0x07000A01); upper limit. The final position of the adjustment procedure is the position specified by <LowValue>.

When the <LowValue> value is outside of the permissible position range of the axis (specified by ***Position Range Limit min***, ID 0x07000000, and ***Position Range Limit max***, ID 0x07000001), it is set to the respective limit value.

When "NaN" is entered for <LowValue>, the value of the ***autozero Low Value*** parameter (ID 0x07000A00) is used.

The adjustment procedure changes the value of the ***autozero Result*** parameter (ID 0x07000A03) in the volatile memory. This value is interpreted as the force in N and is set as the control value of the axis when the servo mode is switched off.

When a force sensor is allocated to the axis via the ***Input Channel For Force Feedback*** parameter (ID 0x07000400), the adjustment procedure also changes the value of the ***Sensor Mech. Correction 1*** parameter (ID 0x02000200) for the input signal channel of the force sensor. The parameter value is changed in the volatile memory.

If the currently valid values of the above-mentioned parameters are to be maintained when the C-413 is switched off or rebooted, they have to be saved with WPA (p. 229); refer also to "Adapting Settings" (p. 257).

To get write access for parameters, it can be necessary to switch to command level 1 with CCL (p. 157).

Refer also to "Autozero Procedure for Compensating the Weight Force" (p. 49)

Example 1: The value of the ***Autozero Low Value*** parameter saved for axis 1 in the controller is checked.

Send: SEP? 1 0x07000A00

Receive: 1 0x7000a00=0.000000e+00

The autozero procedure for axis 1 is started with the value of the ***Autozero Low Voltage*** parameter.

Send: ATZ 1 NaN

The success of the autozero procedure for axis 1 is checked (procedure was successful).

Send: ATZ? 1

Receive: 1

Example 2: The autozero procedure for axis 1 is started, whereby the control value must be set at position 15 (mm) so that the force takes on the value zero (N).

Send: ATZ 1 15.0

The success of the autozero procedure for axis 1 is checked (procedure was not successful).

Send: ATZ? 1

Receive: 0

ATZ? (Get State Of Automatic Zero Point Calibration)

Description: Queries success of the automatic zero point adjustment (see ATZ (p. 154) for details).

Format: ATZ? [{<AxisID>}]

Arguments <AxisID> is one axis of the controller

Response: {<AxisID>"="<uint> LF}

where

<uint> indicates whether the automatic zero point adjustment of the specified axis was successful (= 1) or not (= 0).

Troubleshooting: Illegal axis identifier

CAV? (Get Current Value Of Controlled Variable)

Description: Queries the current value of the variable that is controlled in the selected control mode (to select the control mode, see CMO (p. 159)).

When the control variable is the position, CAV? corresponds to a position query with POS? (p. 193).

Format: CAV? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<float> LF}

where

	<float> is the current value of the control variable in physical units. The interpretation of the value and thus its physical unit depend on the control mode that is selected for the axis.
Troubleshooting:	Illegal axis identifier

CCL (Set Command Level)

Description:	Changes the active "command level" and therefore determines the availability of commands and write access to system parameters.
Format:	CCL <Level> [<PSWD>]
Arguments:	<Level> is a command level of the controller <PSWD> is the password required for changing to the appropriate command level The following command levels and passwords apply: Level = 0 is the default setting, all commands provided for "normal" users are available, read access to all parameters, no password required. Level = 1 adds additional commands and write access to level-1 parameters (commands and parameters from level 0 are included). The password required is "advanced". Level > 1 is only intended for PI service personnel. Users cannot change to a level > 1. Contact the customer service department (p. 287) if you have problems with the parameters for command level 2 or higher.
Response:	none
Troubleshooting:	Invalid password
Notes:	With C-413, the command levels only determine the write permission for the parameters. The availability of the commands of the C-413 is independent of the active command level. HPA? (p. 184) lists the parameters including the information on which command level allows write access to them. Exception: The AOS (p. 151), CMO (p. 159), WTR (p. 231) and WOS (p. 227) commands can be used to change parameters on command level 0 in the volatile memory, although the HPA? response indicates a higher command level for the write access. Refer to "Adapting Settings" (p. 257) for further information on changing parameters.

The active command level is always level 0 after switching the controller on or rebooting.

CCL? (Get Command Level)

Description:	Get the active "command level".
Format:	CCL?
Arguments:	none
Response:	<Level> is the currently active command level; uint.
Notes:	<Level> should be 0 or 1. <Level> = 0 is the default setting, write access is specified for level 0 parameters, read access is specified for all parameters Exception: The AOS (p. 151), CMO (p. 159), WTR (p. 231) and WOS (p. 227) commands can be used to change parameters on command level 0 in the volatile memory, although the HPA? response (p. 184) indicates a higher command level for the write access. <Level> = 1 allows write access for level 1 parameters (parameters from level 0 are included).

CCV? (Get Control Value)

Description:	Queries the currently valid control value of the axis.
Format:	CCV? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response:	{<AxisID>"="<float> LF} where <float> is the currently valid control value.
Troubleshooting:	Illegal axis identifier
Notes:	The control value corresponds to the force in N to be generated. The control value results from the following components: Closed-loop operation: ▪ Result of the servo algorithm ▪ Correction via notch filter Open-loop operation: ▪ Immediately after the servo mode is switched off: Specification by the value of the ***AutoZero Result***

parameter (ID 0x07000A03, see ATZ (p. 154))

- Specification by control source, e.g., motion commands (SVA (p. 205), SVR (p. 208), IMP (p. 187), STE (p. 202)), analog control input or wave generator
- Correction via slew rate limitation
- Correction via notch filter

Refer to "Generating Control Values" (p. 24) for further information.

CMN? (Get Minimum Commandable Closed-Loop Target)

Description: Queries the minimum target value that can be commanded in the selected control mode (to select the control mode, see CMO (p. 159)).

When the control variable is the position, CMN? corresponds to querying the minimum commandable position with TMN? (p. 210).

Format: CMN? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response {<AxisID>"="<float> LF}

where

<float> is the minimum commandable target value in physical units. The physical unit and thus the interpretation of the value depend on the control mode that is selected for the axis.

Note: The minimum commandable target value is defined according to the selected control mode as follows:

- Control variable is the position: Value of the ***Position Range Limit min*** parameter (ID 0x07000000)
- Control variable is the velocity: Value of the ***Profile Generator Maximum Velocity*** parameter (ID 0x06010400) with negative sign
- Control variable is the force: Value of the ***Force Range Limit min*** parameter (ID 0x07000005)

CMO (Set Closed-Loop Control Mode)

Description: Selects the control mode for closed-loop operation.

The selection of the control mode determines the control variable. The target value for the control variable can be set with CTV (p. 169) and CTR (p. 168). The current value of the control variable can be queried with CAV? (p. 156).

Format: CMO {<AxisID> <CtrlMode>}

Arguments: <AxisID> is one axis of the controller

<CtrlMode> is the ID of the control mode to be selected; for supported control modes and their IDs, see below.

Response: None

Troubleshooting: Illegal axis identifier

Notes: CMO sets the value of the ***Closed-Loop Control Mode*** parameter (ID 0x07030100) in the volatile memory. Required command level: 0. (Setting the parameter value with SPA (p. 199) requires switching to command level 1 with CCL (p. 157).)

If the settings made with CMO are to be preserved when the C-413 is switched off or rebooted, they have to be saved with WPA (p. 229); refer also to "Adapting Settings" (p. 257).

The ***Available Closed-Loop Control Modes*** parameter (ID 0x07030101) limits the control modes supported by C-413 to control modes that can actually be selected. This is to prevent the selection of a control mode for which the servo control parameters of the C-413 are not adapted.

In the following cases, it is not permissible to change the selected control mode:

- The wave generator is running for the axis.
- An input signal channel is used as the control source for the axis.

When the control mode is changed in closed-loop operation, the target value for the control variable is set as follows to prevent jumps in the mechanics:

- The new control variable is the position: The target value is set to the current value of the position.
- The new control variable is the velocity or the force: The target value is set to zero.

Supported control modes: The following table lists the control modes that are supported by the C-413. The default settings are marked:

- Bold: Selected control mode
- Gray background: Selectable control modes

ID	Short designation	Control mode	Control variable	Supported motion commands in closed-loop operation
1	PID_Pos	Direct PID position control	Position	MOV, MVR, CTV, CTR, STE, IMP
6	PID_Vel	Direct PID velocity control	Velocity	VEL, CTV, CTR, STE, IMP
7	**PID_Pos_Vel**	**PID position control with velocity control**	**Position**	**MOV, MVR, CTV, CTR, STE, IMP**
8	PID_Force	Direct PID force control	Force	CTV, CTR, IMP, STE
9	PID_Force_Pos	PID force control with position control	Force	CTV, CTR, IMP, STE
10	PID_Force_Pos_Vel	PID force control with position control and velocity control	Force	CTV, CTR, IMP, STE
11	PID_Force_Vel	PID force control with velocity control	Force	CTV, CTR, IMP, STE

CMO? (Get Closed-Loop Control Mode)

Description: Queries the selected control mode for closed-loop operation (value of the ***Closed-Loop Control Mode*** parameter in the volatile memory (ID 0x07030100)).

Format: CMO? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<CtrlMode> LF}

where

<CtrlMode> is the ID of the selected control mode. See CMO (p. 159) for supported control modes.

Troubleshooting: Illegal axis identifier

CMX? (Get Maximum Commandable Closed-Loop Target)

Description: Queries the maximum target value that can be commanded in the selected control mode (to select the control mode, see CMO (p. 159)).

	When the control variable is the position, CMX? corresponds to querying the maximum commandable position with TMX? (p. 210).
Format:	CMX? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response	{<AxisID>"="<float> LF} where <float> is the maximum commandable target value in physical units. The physical unit and thus the interpretation of the value depend on the control mode that is selected for the axis.
Note:	The maximum commandable target value is defined according to the selected control mode as follows: ▪ Control variable is the position: Value of the ***Position Range Limit max*** parameter (ID 0x07000001) ▪ Control variable is the velocity: Value of the ***Profile Generator Maximum Velocity*** parameter (ID 0x06010400) with positive sign ▪ Control variable is the force: Value of the ***Force Range Limit max*** parameter (ID 0x07000006)

CST? (Get Assignment Of Stages To Axes)

Description:	Returns the name of the connected positioner type for the queried axis.
Format:	CST? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	{<AxisID>"="<string> LF} where <string> is the name of the positioner type assigned to the axis.
Notes:	The name is read from the ***Stage Type*** parameter (ID 0x0F000100) of the input signal channel that is allocated to the axis via the input matrix (p. 16). The C-413 reads the value of the ***Stage Type*** parameter from the ID chip (p. 53) of the connected mechanics when it is switched on or rebooted.

CSV? (Get Current Syntax Version)

Description:	Queries the GCS syntax version used in the firmware.
Format:	CSV?
Arguments:	None
Response:	The current GCS syntax version
Notes:	The response is always 2.0 (for GCS 2.0).

CTI (Set Configuration Of Trigger Input)

Description: Configures the trigger input for the specified digital input line.

Format: CTI {<TrigInID> <CTIPam> <Value>}

Arguments: <TrigInID> is one digital input line of the controller; see below for further information.

<CTIPam> is the ID of the CTI parameter in decimal format, see below for available IDs.

<Value> is the value that the CTI parameter is set to, see below.

Response: None

Notes: The trigger configuration must be activated with TRI (p. 211).

The CTI settings are lost when the C-413 is switched off or rebooted.

Available input lines and configuration options: <TrigInID> corresponds to digital input lines 1 to 4, IDs = 1 to 4; see "I/O" (p. 295).

<CTIPam> parameter IDs available for C-413:
1 = TriggerType
3 = TriggerMode
7 = Polarity
13 = WaveGenerator

<Value> available for the appropriate <CTIPam> ID:

for TriggerType:

- 0 = Edge triggered; triggering upon state transition of the digital input line. The activating state transition can be low --> high or high --> low (depends on the signal polarity set (CTIPam 7)).
- 1 = Level triggered; triggering when the digital input line is in an active state (high or low; depends on the signal polarity set (CTIPam 7)).

for TriggerMode:

- 0 = No triggering
- 2 = DataRecorder;
  The digital input line triggers a recording by the data recorder. With DRT (p. 176) the "External trigger" trigger option must be set, and the digital input line selected with DRT must match <TrigInID>.
  It is not possible to trigger a recording again until after the recording in progress has ended (i.e. when the data recorder tables are full) and requires "External trigger" to be set again with DRT.
  Refer to "Data Recorder" (p. 83) for further information.
- 4 = WaveGenerator;
  The digital input line starts/interrupts the wave generator output. For the selected wave generators (CTIPam ID 13), the start mode "Start via external trigger signal" (bit 1) must be set with WGO (p. 224).
  The wave generator output depends on the selected trigger type (CTIPam 1):
  Edge triggered: Each activating state transition of the digital input line triggers the output of a point in the wave table. When an output rate > 1 is set with WTR (p. 231), the corresponding number of activating state transitions is required to output a point.
  Level triggered: When the digital input line is in the active state, the wave generator outputs the points of the wave table. When the digital input line is in the non-active state, the wave generator output is interrupted.
  Regardless of the selected trigger type, the number of output cycles of the waveform can be limited with WGC (p. 223).
  Refer to "Wave Generator" (p. 120) for further information.

for polarity: Sets the signal polarity for the digital input line

- 0 = active low
- 1 = active high

for WaveGenerator: Bit-coded specification of the wave generators that are to be connected with the digital input line in WaveGenerator trigger mode. The value is to be specified as the sum of the following codes of the wave generators:
1 = wave generator 1
2 = wave generator 2
4 = wave generator 3

...
The C-413 has 2 wave generators.

Refer to "Digital Input Signals" (p. 90) for application examples and further information.

CTI? (Get Configuration Of Trigger Input)

Description:	Queries the values set for specified trigger input lines and parameters.
Format:	CTI? [{<TrigInID> <CTIPam>}]
Arguments:	<TrigInID> is one digital input line of the controller, see CTI. <CTIPam>: parameter ID; see CTI. If all arguments are left out, the response contains the values for all parameters and all input lines.
Response:	{<TrigInID> <CTIPam>"="<Value> LF} For <Value> see CTI.

CTO (Set Configuration Of Trigger Output)

Description:	Configures the trigger output conditions for the specified digital output line.
Format:	CTO {<TrigOutID> <CTOPam> <Value>}
Arguments:	<TrigOutID> is one digital output line of the controller, see below for details. <CTOPam> is the CTO parameter ID in decimal format, see below for the available IDs. <Value> is the value that the CTO parameter is set to, see below.
Response:	None
Notes:	The trigger output conditions will become active when activated with TRO (p. 212). Do not use DIO (p. 171) on digital output lines for which the trigger output is activated with TRO. The trigger output takes place independently of the selected control mode (see CMO (p. 159)). The CTO settings are lost when the C-413 is switched off or rebooted.

Output lines and trigger conditions available:	<TrigOutID> corresponds to digital output lines 1 to 5, IDs = 1 to 5; see "I/O" (p. 295).

<CTOPam> parameter IDs available for the C-413:

1 = TriggerStep
2 = Axis
3 = TriggerMode
5 = MinThreshold
6 = MaxThreshold
7 = Polarity
8 = StartThreshold
9 = StopThreshold

<Value> available for the appropriate <CTOPam> ID:

for TriggerStep: Distance

for Axis: The identifier of the axis to be connected to the digital output line.

for TriggerMode:

- 0 = PositionDistance;
  a trigger pulse is written whenever the axis has covered the TriggerStep distance (<CTOPam> ID 1). Optionally, values for StartThreshold and StopThreshold (<CTOPam> IDs 8 and 9) can be defined to activate the trigger output for a limited position range and a certain direction of motion only (negative or positive; Note: If the motion direction is reversed before the axis position has reached the stop threshold, trigger pulses will continue to be generated). When StartThreshold and StopThreshold are set to the same value, they will not be used.
- 2 = OnTarget;
  the on-target state of the selected axis is transferred to the selected digital output line (this state can also be read with the ONT? command).
- 3 = MinMaxThreshold;
  when the position of the selected axis is within the band that is defined by MinThreshold and MaxThreshold (<CTOPam> IDs 5 and 6), the selected digital output line is active.
- 6 = InMotion;
  the selected digital output line is active as long as the selected axis is in motion (the motion state can also be read with commands, e.g. SRG? or #5).

for MinThreshold/MaxThreshold: Position value; used for the MinMaxThreshold trigger mode; both values must be

	set to form a band for Polarity: Sets the signal polarity for the digital output line 0 = Active Low 1 = Active High for StartThreshold/StopThreshold: Position value; can be used for the PositionDistance trigger mode; both thresholds must be set to determine the position range and the direction of motion for the trigger output Refer to "Digital Output Signals" (p. 85) and the lines below for application examples and further details.
Example 1:	A pulse is to be generated on digital output line 1 (ID 1) whenever axis 1 has covered a distance of 0.05 µm. The following parameters must be set: TrigOutID = 1 Axis = 1 TriggerMode = 0 TriggerStep = 0.05 Send: `CTO 1 2 1` Send: `CTO 1 3 0` Send: `CTO 1 1 0.00005`

Example 2:	In this example, digital output line 1 is to be set from low to high when axis 2 starts to move. The following parameters must be set:
	TrigOutID = 1
	Axis = 2
	TriggerMode = 6
	Polarity = Active High
	So you have to send:
	`CTO 1 2 2`
	`CTO 1 3 6`
	`CTO 1 7 1`

CTO? (Get Configuration Of Trigger Output)

Description:	Queries the values set for specified trigger output lines and parameters.
Format:	CTO? [{<TrigOutID> <CTOPam>}]
Arguments:	<TrigOutID>: is a digital output line of the controller; see CTO.

<CTOPam>: parameter ID; see CTO.

If all arguments are left out, the response contains the values for all parameters and all output lines.

Response: {<TrigOutID> <CTOPam>"="<Value> LF}

For <Value> see CTO.

CTR (Set Target Relative To Current Closed-Loop Target)

Description: Sets a new target value relative to the last valid target value.

Servo mode must be switched on when this command is used (closed-loop operation).

The control variable for which the relative target value is set with CTR depends on the selected control mode (to select the control mode, see CMO (p. 159)).

When the control variable is the position, CTR corresponds to setting the target position with MVR (p. 189).

Format: CTR {<AxisID> <TargetRelative>}

Arguments: <AxisID> is one axis of the controller.

<TargetRelative> is the relative target values in physical units. The sum of the relative target value and the currently valid target value is set as the new absolute target value. The physical unit and thus the interpretation of the value depend on the control mode that is selected for the axis.

Response: None

Notes: An absolute target value for the control variable can be set with CTV (p. 169). The current value of the control variable can be queried with CAV? (p. 156).

The target value must be within the limits for the control variable. Use CMN? (p. 159) and CMX? (p. 161) to get the currently valid limits. The last valid target value for the control variable can be queried with CTV?.

The motion can be stopped by #24 (p. 150), STP (p. 204) and HLT (p. 183).

During a motion, a new motion command resets the target to a new value and the old value may never be reached.

Motion commands such as CTR are not allowed when the analog control input or wave generator output is active for the axis. Refer to "Generating Control Values" (p. 24) for further information.

For information on protecting the connected mechanics

- When the control variable is the velocity or the force: The axis can move to the hard stop at a high velocity.
- When the axis is in the overflow state for more than 60 s (get with OVF? (p. 191)), the C-413 switches off the servo mode for the axis. Possible causes for the occurrence of the overflow state:
  - When the control variable is the position or the velocity: The axis is blocked by an obstacle.
  - When the control variable is the velocity or the force: The axis has reached the hard stop.

CTV (Set Absolute Closed-Loop Target)

Description: Sets a new absolute target value for the specified axis.

Servo mode must be switched on when this command is used (closed-loop operation).

The control variable for which the target value is set with CTV depends on the selected control mode (to select the control mode, see CMO (p. 159)).

When the control variable is the position, CTV corresponds to setting the target position with MOV (p. 188).

When the control variable is the velocity, CTV corresponds to setting the target velocity with VEL (p. 215).

Format: CTV {<AxisID> <TargetAbsolute>}

Arguments: <AxisID> is one axis of the controller.

<TargetAbsolute> is the absolute target value in physical units. The physical unit and therefore the interpretation of the value depend on the control mode selected for the axis.

Response: None

Notes: The last valid target value for the control variable can be queried with CTV?. A relative target value for the control variable can be set with CTR (p. 168). The current value of the control variable can be queried with CAV? (p. 156).

The target value must be within the limits for the control variable. Use CMN? (p. 159) and CMX? (p. 161) to get the current valid limits.

The motion can be stopped by #24 (p. 150), STP (p. 204) and HLT (p. 183).

During a motion, a new motion command resets the target to a new value and the old value may never be reached.

Motion commands such as CTV are not allowed when the analog control input or the wave generator output is active for the axis. Refer to "Generating Control Values" (p. 24) for further information.

For information on protecting the connected mechanics:

- When the control variable is the velocity or the force: The axis can move to the hard stop at a high velocity.
- When the axis is in the overflow state for more than 60 s (get with OVF? (p. 191)), the C-413 switches off the servo mode for the axis. Possible causes for the occurrence of the overflow state:
  - When the control variable is the position or the velocity: The axis is blocked by an obstacle.
  - When the control variable is the velocity or the force: The axis has reached the hard stop.

CTV? (Get Closed-Loop Target)

Description: Queries the currently valid target value for closed-loop operation.

The control variable for which the target value is queried with CTV? depends on the selected control mode (to select the control mode, see CMO (p. 159)).

When the control variable is the position, CTV? corresponds to querying the target position with MOV? (p. 189).

When the control variable is the velocity, CTV? corresponds to querying the target velocity with VEL? (p. 216).

Format: CTV? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<float> LF}

where

	<float> is the currently valid target value for closed-loop operation in physical units. The physical unit and therefore the interpretation of the value depends on the control mode that is selected for the axis.
Troubleshooting:	Illegal axis identifier
Notes:	The target value can be changed by various sources, e.g., by commands that cause motion (for an overview, see CMO (p. 159)), by the wave generator or by an analog input signal. Refer to "Generating Control Values" (p. 24) for further information. CTV? queries the current target value. The actual value of the control variable can be queried with CAV? (p. 156).

DIO (Set Digital Output Line)

Description:	Switches the specified digital output line(s) to specified state(s). Use TIO? (p. 209) to get the number of installed digital I/O lines.
Format:	DIO {<DIOID> <OutputOn>}
Arguments:	<DIOID> is one digital output line of the controller, see below for details. <OutputOn> is the state of the digital output line, see below for details.
Response:	none
Notes:	Using the DIO command, digital output lines 1 to 5, which are located on the **I/O** panel plug (p. 295), can be activated/deactivated. The <DIOID> identifiers to be used for the lines are 1 to 5. If <OutputOn>=1 the line is set to HIGH/ON; if <OutputOn>=0 it is set to LOW/OFF. Do not use DIO on output lines where the trigger output is activated with TRO (p. 212).

DIO? (Get Digital Input Lines)

Description:	Queries the states of the specified digital input lines.

	Use TIO? (p. 209) to query the number of available digital I/O lines.
Format:	DIO? [{<DIOID>}]
Arguments:	<DIOID> is the identifier of the digital input line, see below for details.
Response:	{<DIOID>"="<InputOn> LF} where <InputOn> specifies the state of the digital input line, see below for details.
Notes:	The DIO? command can be used to directly read digital input lines 1 to 4, which are located on the **I/O** panel plug (p. 295). The <DIOID> identifiers to be used for the lines are 1 to 4. If the identifier is omitted, all lines are queried. If <InputOn>=0, the digital input signal is LOW/OFF; if <InputOn>=1, the digital input signal is HIGH/ON.

DRC (Set Data Recorder Configuration)

Description:	Determines the data source to be used and the type of data to be recorded (record option) for the data recorder table specified.
Format:	DRC {<RecTableID> <Source> <RecOption>}
Arguments:	<RecTableID> is one data recorder table of the controller, see below. <Source> is the ID of the data source, for example, an axis or channel of the controller. The required source depends on the selected record option. <RecOption> is the type of data to be recorded (record option). Refer to the following list of available record options and the corresponding data sources for details
Response:	None
Notes:	The number of available data recorder tables can be read with TNR? (p. 210) The response is the value of the ***Data Recorder Channel Number*** parameter (ID 0x16000300). With HDR? (p. 181) you will obtain a list of available record options and information on additional parameters and commands concerned with data recording.

Refer to "Data Recorder" (p. 83) for further information.

Record options for the appropriate data sources:	<Source>	<RecOption>
	Axis	0 = Nothing is recorded 1 = Target Position of axis (i.e. target position value in closed-loop operation), corresponds to the MOV? response 2 = Current Position of axis, corresponds to the POS? response 3 = Position Error of axis 14 = Open Loop Control of axis (i.e. the control value for open-loop operation), corresponds to the SVA? response 15 = PID Control Output of axis (i.e. result of the servo algorithm for the selected control mode, before correction by the notch filter) 22 = Slew-Rate Limited Target Position of axis (in closed-loop operation), target position after slew rate limitation 28 = Closed-Loop Target Value of axis (i.e. target value in closed-loop operation), corresponds to the CTV? response 30 = Current Value of axis (i.e., current value of the variable controlled in the selected control mode), corresponds to the CAV? response 31 = Control Value of axis (i.e. currently valid control value), corresponds to the CCV? response
	Output Signal Channel	16 = Output Value of output signal channel (after the axis-to-output signal channel transformation; can be interpreted as the output current for the drive or the position, force or velocity of an axis (corresponding voltage value) depending on the output type definition), corresponds to the VOL? response 33 = I2T Value, current I^2t value for the output current to the drive

Input Signal Channel	18 = Input signal channel, after sensor filtering
	19 = Input signal channel, after sensor electronics linearization
	20 = Input signal channel, after sensor mechanics linearization, corresponds to the TSP? response

For more information, see "Functional Principles" (p. 11).

Example: Send `DRC 4 1 2`

The current position of axis 1 is recorded in data recorder table 4 with the next recording.

DRC? (Get Data Recorder Configuration)

Description: Queries the settings for the data to be recorded.

Format: DRC? [{<RecTableID>}]

Arguments: <RecTableID>: is a data recorder table of the controller; if this entry is not specified, the response will contain the settings for all tables.

Response: The current DRC settings:

{<RecTableID>"="<Source> <RecOption> LF}

where

<Source>: is the data source, for example, an axis or a channel of the controller. The source type depends on the record option.

<RecOption>: is the type of data to be recorded (record option).

The available record options can be queried with HDR? (p. 181).

DRL? (Get Number of Recorded Points)

Description: Reads the number of points comprised by the last recording.

Format: DRL? [{<RecTableID>}]

Arguments: <RecTableID> is one data recorder table of the controller

Response: {<RecTableID>"="<uint> LF}

where

<uint> specifies the number of points recorded with the last recording

Notes: The number of points is reset to zero for a data recorder table when changing its configuration with DRC (p. 172).

DRR? (Get Recorded Data Values)

Description: Queries the last recorded data.

Querying can take some time depending on the number of points to be read!

It is possible to read the data while recording is still in progress.

Format: DRR? [<StartPoint> <NumberOfPoints> [{<RecTableID>}]]

Arguments: <StartPoint> is the first point to be read from the data recorder table, starts with index 1.

<NumberOfPoints> is the number of points to be read per table.

<RecTableID> is one data recorder table of the controller.

Response: For the recorded data in GCS array format, refer to the separate manual for the GCS array, SM146E, and the example below.

Notes: If <RecTableID> is omitted, the data from all tables with a nonzero record option is read.

With HDR? (p. 181) you will obtain a list of available record options and information on additional parameters and commands concerning data recording.

Refer to the description of the DRC command (p. 172) as well as "Data Recorder" (p. 83) for further information.

Example:

```
rtr?
1
drr? 1 20 1
# REM Dataset sent by C-413.2GA, Serial
Number:0114009224
# REM Content: 20 Record Table Data of
```

```
Record Table 1 from Start Point 1
# TYPE = 1
# SEPARATOR = 9
# DIM = 1
# SAMPLE_TIME = 2.02666e-4
# NDATA = 20
# NAME0 = Real Position of Axis 1
# END_HEADER
-4.800006
-4.7999836
-4.799988
-4.7999788
-4.799982
-4.8000204
-4.8000056
-4.7999568
-4.7999852
-4.799994
-4.799976
-4.7999716
-4.7999648
-4.7999436
-4.7999508
-4.79995
-4.7999356
-4.7999228
-4.7999084
-4.7998692
```

DRT (Set Data Recorder Trigger Source)

Description:	Defines a trigger source for the specified data recorder table.
Format:	DRT <RecTableID> <TriggerSource> <Value>
Arguments:	<RecTableID> is one data recorder table of the controller. See below for details. <TriggerSource> ID of the trigger source, see below for a list of available options. <Value> depends on the trigger source, can be a dummy, see below.
Response:	none

Notes: Regardless of the data recorder table selected with <RecTableID>, the specified trigger source is always set for all data recorder tables. <RecTableID> can also have the value zero (= "all data recorder tables").

Regardless of the trigger option set, the data recording is always triggered in the following cases:

- Starting a step response measurement with STE (p. 202)
- Starting an impulse response measurement with IMP (p. 187)
- Starting the wave generator with WGO (p. 224), bit 0
- When the wave generator is running: Start of the data recording with WGR (p. 227)

With HDR? (p. 181) you will obtain a list of available record options and information on additional parameters and commands concerning data recording.

Refer to the description of the DRC command (p. 172) as well as "Data Recorder" (p. 83) for further information.

Available trigger options: 0 = default setting; data recording is triggered with STE, IMP, WGO bit 0 and WGR; <Value> must be a dummy.

3 = external trigger; data recording is started with the digital input line whose ID is specified by <Value>. <Value> can also have the value zero (= "all digital input lines"). The selected digital input line must be additionally configured with the CTI (p. 163) and TRI (p. 211) commands. The trigger option is reset to "default setting" after execution.

4 = immediately, starts data recording immediately and sets trigger option back to "default setting" after execution; <Value> must be a dummy.

DRT? (Get Data Recorder Trigger Source)

Description: Queries the trigger source for the data recorder tables.

Format: DRT? [{<RecTableID>}]

Arguments: <RecTableID> is one data recorder table of the controller.

Response: {<RecTableID>"="<TriggerSource> <Value> LF}

where

<TriggerSource> is the identifier of the trigger source.

<Value> depends on the trigger source.

Further information can be found in the description of the DRT command (p. 176).

ERR? (Get Error Number)

Description:	Get error code <int> of the last occurred error and reset the error to 0. Only the last error is buffered. You should therefore call ERR? after each command. The error codes and their descriptions are listed in "Error Codes" (p. 233).
Format:	ERR?
Arguments:	None
Response:	The error code of the last error that occurred (integer).
Troubleshooting:	Communication breakdown
Notes:	In the case of simultaneous access to the controller by several instances, the error code is only returned to the first instance that sent the ERR? command. Because the error is reset to 0 by the query, the error is not visible for any further querying instance. ➢ If possible, access the controller with one instance only. ➢ If incorrect system behavior does not cause the controller to send an error code, check whether the error code is queried regularly in the background by a macro, script or the PC software (e.g., PIMikroMove). If the cause of an error continues, the corresponding error code is immediately set again after a query with ERR?.

FRF (Fast Reference Move To Reference Switch)

Description:	Starts a referencing move. Moves the specified axis to the reference switch and sets the current position to a defined value. See below for details. If multiple axes are specified in the command, they are started simultaneously.
Format:	FRF [{<AxisID>}]

Arguments:	<AxisID> is a controller's axis, all axes are affected if not specified.
Response:	None
Troubleshooting:	Illegal axis identifier

Notes: Sequence of the referencing move:

1. The axis moves to the reference switch.
2. At the reference switch, the value of the ***Sensor Mech. Correction 1*** parameter (ID 0x02000200) is set in the volatile memory to an offset value from the ID chip of the sensor.
3. The value of the ***Sensor Mech. Correction 1*** parameter is set as the new current position of the axis.
4. The referencing move ends at the zero position of the axis. The value of the ***Sensor Mech. Correction 1*** parameter determines the behavior:
   - When the parameter value is zero: The axis stays at the reference switch.
   - When the parameter value is not zero: The axis moves from the reference switch to the new zero position.

The ***Sensor Mech. Correction 1*** parameter of the input signal channel that is allocated to the axis via the input matrix (p. 16) is used .

If the referencing move was successful, motion to absolute target values will be possible in closed-loop operation afterwards.

Motion can be stopped by #24 (p. 150), STP (p. 204) and HLT (p. 183).

Use FRF? (p. 180) to check whether the referencing move was successful.

The ***Power Up Reference Move Enable*** parameter (ID 0x07000806) can be used to configure the C-413 so that the referencing move is automatically performed after switching on or rebooting.

Refer to "Referencing" (p. 46) for further information.

FRF? (Get Referencing Result)

Description: Queries whether the specified axis is referenced or not.

Format: FRF? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<uint> LF}

where

<uint> indicates whether the axis has been successfully referenced (=1) or not (=0).

Troubleshooting: Illegal axis identifier

Notes: An axis is considered as "referenced" when the current position value is set to a known position. This is the case when a referencing move was done successfully with FRF (p. 178) or when the position was set directly with POS (p. 192) (depending on the referencing method set with RON (p. 194)).

GWD? (Get Wave Table Data)

Description: Queries waveform for specified wave table.

Format: GWD? [<StartPoint> <NumberOfPoints> [{<WaveTableID>}]]

Arguments: <StartPoint> is the start point in the wave table, begins with index 1

<NumberOfPoints> is the number of points to be read per table

<WaveTableID> is one wave table of the controller

Response: The wave table contents (waveform) in GCS array format (refer to the separate manual for the GCS array, SM 146E, and the example below)

Notes: Depending on the waveform definition with WAV (p. 217), the wave tables can have different lengths. Due to the definition of the PI General Command Set response format, wave tables with different lengths can only be queried with the following syntax:

GWD? <StartPoint> <NumberOfPoints> {<WaveTableID>}

The response to GWD? does not contain any offset to the wave generator output set with WOS (p. 227).

Example:

```
gwd? 1 20 1 2
# REM Dataset sent by C-413.2GA, Serial
```

```
Number:0114009224
# REM Content: 20 Wave Table Data of
Wave Table 1 and 2 from Start Point 1
# TYPE = 1
# SEPARATOR = 9
# DIM = 2
# SAMPLE_TIME0 = 2.02666e-4
# SAMPLE_TIME1 = 2.02666e-4
# NDATA = 20
# NAME0 = Data Wave Table 1
# NAME1 = Data Wave Table 2
# END_HEADER
0.0   0.0
0.0   0.0
4.9352648e-5    1.1849403e-4
1.9741058e-4    4.7373772e-4
4.439354e-4     1.0658503e-3
7.8940392e-4    1.894474e-3
1.2335777e-3    2.9599668e-3
1.7764568e-3    4.2617324e-3
2.4175644e-3    5.7996512e-3
3.1576158e-3    7.5734856e-3
3.996134e-3     9.5829968e-3
4.9331188e-3    1.1827946e-2
5.9688092e-3    1.4307738e-2
7.102728e-3     1.7022372e-2
8.3351136e-3    1.9971014e-2
9.6659664e-3    2.3153544e-2
1.1095047e-2    2.6569128e-2
1.2622357e-2    3.0217408e-2
1.4248133e-2    3.4097792e-2
1.5971661e-2    3.8209676e-2
```

HDR? (Get All Data Recorder Options)

Description:	Lists a help string which contains all information available on data recording (record options and trigger options, information on additional parameters and commands concerning data recording).
Format:	HDR?
Arguments:	None
Response	#RecordOptions {<RecOption>"="<DescriptionString>[of <Channel>]}

#TriggerOptions
[{<TriggerOption>"="<DescriptionString>}]

#Parameters to be set with SPA
[{<ParameterID>"="<DescriptionString>}]

#Additional information
[{<Command description>"("<Command>")"}]

#Sources for Record Options
[{<RecOption>"="<Source>}]

end of help

Example: For the C-413, the response to HDR? reads as follows:

```
#RecordOptions
0=Nothing is recorded
1=Target Position of Axis
2=Current Position of Axis
3=Position Error of Axis
14=Open-Loop Control Value of Axis
15=PID Control Output Value of Axis
16=Output Value of Output Signal Channel
18=Filtered Sensor Value of Input Signal
Channel
19=Sensor Elec. Linearized Value of
Input Signal Channel
20=Sensor Mech. Linearized Value of
Input Signal Channel
22=Target Position of Axis, Slew Rate
Limited
28=Closed-Loop Target Value of Axis
30=Current Value of Controlled Variable
of Axis
31=Control Value of Axis
33=I2T Value
#TriggerOptions
0=Default
3=External Trigger, reset trigger after
execution
4=Immediately, reset trigger after
execution
#Parameters to be set with SPA
0x16000000=Data Recorder Table Rate
0x16000300=Data Recorder Channel Number
#Additional information
```

```
Set Data Recorder Configuration (DRC
{<RecTableID> <Source> <RecOption>})
Get Data Recorder Configuration (DRC?
[{<RecTableID>}])
Get Number Of Recorded Points (DRL?
[{<RecTableID>}])
Get Recorded Data Values (DRR?
[<StartPoint> [<NumberOfPoints>
[{<RecTableID>}]]])
Set Data Recorder Trigger Source (DRT
<RecTableID> <TriggerSource> <Value>)
Get Data Recorder Trigger Source (DRT?
[{<RecTableID>}])
Set Data Recorder Table Rate (RTR
<RecordTableRate>)
Get Data Recorder Table Rate (RTR?)
Get Number Record Tables (TNR?)
end of help
```

Note: TriggerOptions = 0 (default) means that data recording is triggered by:

- Start of a step response measurement with STE (p. 202)
- Start of an impulse response measurement with IMP (p. 187)
- Start of the wave generator with WGO (p. 224), bit 0
- When the wave generator is running: Start of the data recording with WGR (p. 227)

HLP? (Get List Of Available Commands)

Description:	Lists a help string which contains all commands available.
Format:	HLP?
Arguments:	none
Response:	List of commands available
Troubleshooting:	Communication breakdown

HLT (Halt Motion Smoothly)

Description: Stops the motion of specified axes smoothly. See the notes below for further details.

Error code 10 is set.

#24 (p. 150) and STP (p. 204) in contrast abort current

	motion as fast as possible for the controller without taking care of maximum velocity and acceleration.
Format:	HLT [{<AxisID>}]
Arguments:	<AxisID>: is one axis of the controller, if left out, all axes are stopped
Response:	none
Troubleshooting:	Illegal axis identifier
Notes:	HLT decelerates and stops motion.

HLT stops all motion caused by motion commands (CTV (p. 169), CTR (p. 168), MOV (p. 188), MVR (p. 189), STE (p. 202), IMP (p. 187), VEL (p. 215), SVA (p. 205), SVR (p. 208)), and commands for referencing (FRF (p. 178)).

The target value for the stopped axes in closed-loop operation is set as follows:

- The control variable is the position: The target value is set to the current value of the position.
- The control variable is the velocity or the force: The target value is set to zero.

The control values for the stopped axes in open-loop operation are each set to the value of the ***AutoZero Result*** parameter (ID 0x07000A03, see ATZ (p. 154)).

HPA? (Get List Of Available Parameters)

Description:	Responds with a help string that contains all available parameters with short descriptions. Refer to "Parameter Overview" (p. 267) for further information.
Format:	HPA?
Arguments:	None
Response	{<PamID>"="<string> LF}

where

<PamID> is the ID of one parameter, hexadecimal format

<string> is a string which describes the corresponding parameter.
The string has following format:

<CmdLevel>TAB<MaxItem>TAB<DataType>TAB<FunctionGroupDescription>TAB<ParameterDescription>[{TAB<PossibleValue>"="<ValueDescription>}]

where

<CmdLevel> is the command level that allows write access to the parameter value.

<MaxItem> is the maximum number of elements of the same type that are affected by the parameter. In the case of the C-413, an "element" is an axis, a channel or the entire system.

<DataType> is the data type of the parameter value; it can be INT, FLOAT or CHAR.

<FunctionGroupDescription> is the name of the function group belonging to the parameter
<ParameterDescription> is the parameter name.

<PossibleValue> is a value from the permissible data range.

<ValueDescription> is the meaning of the corresponding value.
The parameters listed with HPA? can be changed and/or saved using the following commands:

SPA (p. 199) influences the parameter settings in volatile memory (RAM).

WPA (p. 229) copies parameter settings from volatile to nonvolatile memory.

SEP (p. 197) writes parameter settings directly into nonvolatile memory (without changing settings in volatile memory).

RPA (p. 195) resets volatile memory to the values from nonvolatile memory.

HPV? (Get Parameter Value Description)

Description:	Responds with a help string that contains possible parameter values. Use HPA? instead to get a help string that contains all available parameters with short descriptions.
Format:	HPV?
Arguments:	None
Response:	<string> has the following format:

"#Possible parameter values are:
{<PamID> <ItemID> "=" <ListType>
[{TAB <PossibleValue> "=" <ValueDescription>}] }
#CCL levels are:
{<PamID> <ItemID> "="<CmdLevel> }
#HPA_Category enabled
end of help"

where

<PamID> is the ID of one parameter, hexadecimal format

<ItemID> is one item (axis, channel, whole system) of the controller, if item=0 the description applies to all items

<ListType> determines how the possible parameter values listed in the string have to be interpreted:
- 0 = parameter not applicable for this item
- 1 = enumeration
- 2 = min/max

<PossibleValue> is a value from the permissible data range

<ValueDescription> is the meaning of the corresponding value

Some parameters are write protected (by a command level > 1) for certain items. These parameters are listed below the "#CCL levels are" line.

<CmdLevel> is the command level that allows write access to the parameter value.

The "#HPA_Category enabled" line is evaluated by the PC software for display purposes.

Notes: The response to HPV? is empty when all required information is already contained in the response to HPA?.

IDN? (Get Device Identification)

Description: Reports the device identity number. Is identical in function with the *IDN? command (p. 151).

Format: IDN?

Arguments: None

Response: One-line string terminated by line feed with controller name, serial number and firmware version, see *IDN? for an example.

IMP (Start Impulse and Response Measurement)

Description:	Starts an impulse and records the impulse response for the specified axis. The data recorder configuration, i.e., the assignment of data sources and record options to the recorder tables, can be set with DRC (p. 172). The recorded data can be read with the DRR? command (p. 175).
Format:	IMP <AxisID> <Amplitude>
Arguments:	<AxisID> is one axis of the controller <Amplitude> is the height of the impulse. See below for details.
Response:	None
Troubleshooting:	The target value (closed-loop operation) or control value (open-loop operation) resulting from the specified impulse height is out of limits: ▪ Open-loop operation: The limitation of the control value results from the parameters 0x07000005 and 0x07000006. ▪ Closed-loop operation: Use CMN? (p. 159) and CMX? (p. 161) to get the currently valid limits. Motion commands such as IMP are not allowed when the analog control input or wave generator output is active for the axis. Refer to "Generating Control Values" (p. 24) for further information.
Notes:	An "impulse" consists of a relative motion with the specified amplitude, followed by an equally large motion in the opposite direction. How the value for <Amplitude> is interpreted depends on the current servo mode: ▪ Closed-loop operation: <Amplitude> is a relative target value in physical units. The control variable, for which the relative target value is set with IMP, depends on the selected control mode (selection of the control mode, see CMO (p. 159)). ▪ Open-loop operation: <Amplitude> is a relative control value. The control value corresponds to the force in N

to be generated.

LIM? (Indicate Limit Switches)

Description:	Queries whether axes have limit switches.
Format:	LIM? [{<AxisID>}]
Arguments:	<AxisID>: is one axis of the controller
Response:	{<AxisID>"="<uint> LF} where <uint> indicates whether the axis has limit switches (=1) or not (=0).
Troubleshooting:	Illegal axis identifier
Notes:	The C-413 does not have any input lines for limit switch signals and only supports the LIM? command for compatibility reasons.

MOV (Set Target Position)

Description:	Sets an absolute target position for the specified axis.
Format:	MOV {<AxisID> <Position>}
Arguments:	<AxisID> is one axis of the controller. <Position> is the absolute target position in physical units.
Response:	none
Notes:	Servo mode must be switched on when this command is used (closed-loop operation). Setting the target position with MOV is only permitted when the control variable is the position (to select the control mode, see CMO (p. 159)). The target position must be within the limits. Use TMN? (p. 210) and TMX? (p. 210) to get the currently valid limits. The motion can be stopped by #24 (p. 150), STP (p. 204) and HLT (p. 183). A new motion command resets the target to a new value during a motion and the old value may never be reached. Motion commands such as MOV are not permitted when the analog control input or wave generator output is active for the axis. Refer to "Generating Control Values" (p. 24)

for further information.

Notes on protecting the mechanics connected:
When the target position cannot be reached because the axis is blocked by an obstacle, the overflow state occurs (get with OVF? (p. 191)). When the axis is in overflow state for more than 60 s, the C-413 switches off the servo mode for the axis.

Example 1: Send: `MOV 1 10`
Note: Axis 1 moves to 10 (target position in mm)

Example 2: Send: `MOV 1 243`
Send: `ERR?`
Receive: `7`
Note: The axis does not move. The error code "7" in the reply to the ERR? command (p. 178) indicates that the target position specified in the motion command is out of limits.

MOV? (Get Target Position)

Description: Returns last valid commanded target position.

Format: MOV? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<float> LF}

where

<float> is the last commanded target position in physical units

Troubleshooting: Illegal axis identifier

Notes: The target position can be changed by various sources, e.g. by commands that cause motion (MOV (p. 188), MVR (p. 189), CTV (p. 169), CTR (p. 168), STE (p. 202), IMP (p. 187)), by the wave generator or by an analog input signal. Refer to "Generating Control Values" (p. 24) for further information.

MOV? queries the commanded position. Use POS? (p. 193) to query the current position.

MVR (Set Target Relative To Current Position)

Description: Moves the specified axis relative to the last commanded target position.

Format:	MVR {<AxisID> <Distance>}
Arguments:	<AxisID> is one axis of the controller. <Distance> specifies the distance that the axis is to move; the sum of the distance and the last commanded target position is set as the new target position (in physical units).
Response:	none
Notes:	Servo mode must be switched on when this command is used (closed-loop operation). Setting the target position with MVR is only permitted when the control variable is the position (to select the control mode, see CMO (p. 159)). The target position must be within the limits. Use TMN? (p. 210) and TMX? (p. 210) to get the currently valid limits, and MOV? (p. 189) to get the current target. Motion can be stopped by #24 (p. 150), STP (p. 204) and HLT (p. 183). A new motion command resets the target to a new value during motion and the old value may never be reached. Motion commands such as MVR are not permitted when the analog control input or wave generator output is active for the axis. Refer to "Generating Control Values" (p. 24) for further information. **Notes on protecting the mechanics connected:** When the target position cannot be reached because the axis is blocked by an obstacle, the overflow state occurs (get with OVF? (p. 191)). When the axis is in overflow state for more than 60 s, the C-413 switches off the servo mode for the axis.
Example:	Send: `MOV 1 0.5` Note: This is an absolute motion. Send: `POS? 1` Receive: `1=0.500000` Send: `MOV? 1` Receive: `1=0.500000` Send: `MVR 1 2` Note: This is a relative motion. Send: `POS? 1` Receive: `1=2.500000` Send: `MVR 1 2000`

Note: New target position of axis 1 would exceed motion range. Command is ignored, i.e., the target position remains unchanged, and the axis does not move.

Send: `MOV? 1`

Receive: `1=2.500000`

Send: `POS? 1`

Receive: `1=2.500000`

ONT? (Get On-Target State)

Description: Queries the on-target state of the specified axis.

If all arguments are left out, queries state of all axes.

Format: ONT? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<uint> LF}

where

<uint> = "1" when the specified axis has reached the target value, otherwise "0".

Troubleshooting: Illegal axis identifier

Notes: The detection of the on-target state is only possible in closed-loop operation (servo mode ON).

The on-target state is influenced by the following settings, depending on the control variable (to select the control mode, see CMO (p. 159)):

- Position: Settling window (parameter 0x07000900) and delay time (parameter 0x07000901)
- Velocity: Settling window (parameter 0x07000902) and delay time (parameter 0x07000903)
- Force: Settling window (parameter 0x07000904) and delay time (parameter 0x07000905)

Refer to "On-Target State" (p. 45) for details.

OVF? (Get Overflow State)

Description: Queries overflow state of specified axis.

If all arguments are left out, queries state of all axes.

Format: OVF? [{<AxisID>}]

Arguments:	<AxisID> is one axis of the controller.
Response:	{<AxisID>"="<uint> LF} where <uint> = "1" when the specified axis is in overflow state, otherwise "0".
Troubleshooting:	Illegal axis identifier
Notes:	The overflow state can only occur in closed-loop operation. The control variable does not reach the target value with the maximum control value in the overflow state. Possible causes for the occurrence of the overflow state: ▪ The axis has not yet been referenced (query with FRF?). ▪ Axis oscillates ▪ When the control variable is the position or the velocity: The axis is blocked by an obstacle. ▪ When the control variable is the velocity or the force: The axis has reached the hard stop. When the axis is in overflow state for more than 60 s, the C-413 switches off the servo mode for the axis.

POS (Set Real Position)

Description:	Sets the current position of the axis (does not cause motion).
Format:	POS {<AxisID> <Position>}
Arguments:	<AxisID> is one axis of the controller. <Position> is the new current position in physical units.
Response:	None
Troubleshooting:	Illegal axis identifier
Notes:	Setting the current position with POS is only possible when referencing method "0" is selected; see RON (p. 194). An axis is considered to be "referenced" when the position was set with POS (refer to "Referencing" (p. 46) for more information). The minimum and maximum commandable positions (TMN? (p. 210), TMX? (p. 210), CMN? (p. 159), CMX? (p. 161)) are not adapted when a position is set with POS. This can result in target positions allowed by the C-413 but

cannot be approached by the hardware. Target positions are also possible that can be reached by the hardware but are denied by the C-413. Furthermore, the zero position can be outside of the physical travel range after using POS.

POS? (Get Real Position)

Description:	Queries the current axis position. If no arguments are specified, the current position of all axes is queried.
Format:	POS? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response:	{<AxisID>"="<float> LF} where <float> is the current axis position in physical units.
Troubleshooting:	Illegal axis identifier

PUN? (Get Axis Unit)

Description:	Queries the current unit of the axis. If all arguments are omitted, the current unit for all axes is queried.
Format:	PUN? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response:	{<AxisID>"="<string> LF} where <string> is the current unit of the axis.
Troubleshooting:	Illegal axis identifier
Note:	PUN? gets the current unit of the variable that is controlled in the selected control mode (to select the control mode, see CMO (p. 159)). The queried unit is only used for display purposes and does not have any influence on the current value of the control variable. The current unit is specified by the following parameter, depending on the control variable: ▪ Position: ***Position Axis Unit*** (parameter 0x07000601) ▪ Velocity: ***Velocity Axis Unit*** (parameter 0x07000603)

- Force: ***Force Axis Unit*** (parameter 0x07000604)

RBT (Reboot System)

Description:	Reboots system. The controller behaves the same as after switching on.
Format:	RBT
Arguments:	none
Response:	none

RON (Set Reference Mode)

Description:	Selects the referencing method for the specified axes
Format:	RON {<AxisID> <ReferenceOn>}
Arguments:	<AxisID> is one axis of the controller. <ReferenceOn> is the referencing method. Can be 0 or 1. 1 is default. See below for details.
Response:	None
Troubleshooting:	Illegal axis identifier
Notes:	<ReferenceOn> = 0: An absolute position value must be assigned to the axis with POS (p. 192). The use of FRF (p. 178) is not permitted. <ReferenceOn> = 1: A referencing move for the axis must be started with FRF. Using POS is not permitted. If the axis has not yet been referenced, relative motion is possible in closed-loop operation with CTR (p. 168), MVR (p. 189), STE (p. 202), and IMP (p. 187) (irrespective of the currently selected referencing method). Refer to "Referencing" (p. 46) for further information.

RON? (Get Reference Mode)

Description:	Queries referencing method of specified axes.
Format:	RON? [{ <AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response:	{<AxisID>"="<ReferenceOn> LF} where

	<ReferenceOn> is the currently selected referencing method for the axis
Troubleshooting:	Illegal axis identifier
Note:	Further information can be found in the description of the RON command (p. 194).

RPA (Reset Volatile Memory Parameters)

Description:	Resets the specified parameter of the specified element. The value from nonvolatile memory is written into volatile memory. Related commands: With HPA? (p. 184) you can obtain a list of the available parameters. SPA (p. 199) influences the parameter settings in volatile memory, WPA (p. 229) writes parameter settings from volatile to nonvolatile memory, and SEP (p. 197) writes parameter settings directly into nonvolatile memory (without changing the settings in volatile memory). See SPA for an example.
Format:	RPA [{<ItemID> <PamID>}]
Arguments:	<ItemID> is the element for resetting a parameter. See below for details. <PamID> is the parameter ID, can be written in hexadecimal or decimal format. See below for details.
Response:	none
Troubleshooting:	Illegal element identifier, wrong parameter ID
Notes:	Only use RPA if you are sure that the C-413 functions correctly with the parameter values from the nonvolatile memory.
Available element IDs and parameter IDs:	An element can be an axis, a channel or the entire system. The element type depends on the parameter. Refer to "Adapting Settings" (p. 257) for further information. Valid parameter IDs are specified in "Parameter Overview" (p. 267).

RTR (Set Record Table Rate)

Description:	Sets the record table rate, i.e., the number of cycles to be used in data recording operations. Settings larger than 1 make it possible to cover longer time periods.

Format: RTR <RecordTableRate>

Arguments: <RecordTableRate> is the record table rate to be used for recording operations (unit: number of cycles), must be an integer value larger than zero.

Response: None

Notes: RTR sets the value of the ***Data Recorder Table Rate*** parameter (ID 0x16000000) in the volatile memory.

The duration of the recording can be calculated as follows:

Rec. Duration = Servo Cycle Time * RTR value * Number of Points

where

Servo Cycle Time for the C-413 is specified by the parameter 0x0E000200 (in seconds)

Number of Points is the length of the data recorder table

If the setting made with RTR is to be maintained when the C-413 is switched off or rebooted, it has to be saved with WPA (p. 229), refer also to "Adapting Settings" (p. 257).

Refer to "Data Recorder" (p. 83) for more information.

RTR? (Get Record Table Rate)

Description: Queries the current record table rate, i.e., the number of cycles used in data recording operations.

Format: RTR?

Arguments: None

Response: <RecordTableRate> is the table rate used for recording operations (unit: number of cycles).

SAI? (Get List Of Current Axis Identifiers)

Description: Queries the axis identifiers.

Refer also to "Commandable Elements" (p. 13).

Format: SAI? [ALL]

Arguments: [ALL] is optional. For controllers that allow deactivating the axis, [ALL] ensures that the response also includes the axes that are "deactivated".

Response:	{<AxisID> LF} <AxisID> is one axis of the controller.
Notes:	Axes can be "deactivated" (p. 52). A deactivated axis is **not** accessible for axis-related commands (e .g. motion commands or queries). With the C-413, deactivated axes are neither in the response to `SAI?` nor in the response to `SAI? ALL`.

SEP (Set Non-Volatile Memory Parameters)

Description:	Sets a parameter of a specified element to a different value in nonvolatile memory, where it becomes the new default. After parameters were set with SEP, you can use RPA (p. 195) to activate them (write them to volatile memory) without controller reboot. **Note that this command is for setting hardware-specific parameters. Wrong values may lead to improper operation or damage of your hardware!** Related commands: HPA? (p. 184) returns a list of the available parameters. SPA (p. 199) writes parameter settings into volatile memory (without changing the settings in nonvolatile memory). WPA (p. 229) writes parameter settings from volatile to nonvolatile memory.
Format:	SEP <Pswd> {<ItemID> <PamID> <PamValue>}
Arguments	<Pswd> is the password for writing to the nonvolatile memory; the default value is "100". <ItemID> is the element for changing a parameter in the nonvolatile memory. See below for details. <PamID> is the parameter identifier, can be written in hexadecimal or decimal format. See below for details. <PamValue> is the value for setting the specified parameter of the specified element.
Response:	None
Troubleshooting:	Illegal element identifier, wrong parameter ID, invalid

	password
Notes:	**Note that the number of write cycles in the nonvolatile memory is limited. Write default settings only if necessary.**
Available element IDs and parameter IDs:	An element can be an axis, a channel or the entire system. The element type depends on the parameter. Refer to "Adapting Settings" (p. 257) for further information. Valid parameter IDs are specified in "Parameter Overview" (p. 267).
Example:	For axis 2 of the C-413, the servo mode is to be automatically switched on after switching on or rebooting. For this purpose, the ***Power Up Servo Enable*** parameter (ID 0x07000800) is set to the value 1 in the nonvolatile memory. The parameter ID is written in hexadecimal format. Send: `SEP 100 2 0x07000800 1`

SEP? (Get Nonvolatile Memory Parameters)

Description:	Queries the value of a parameter of a specified element from nonvolatile memory. With HPA? (p. 184) you can obtain a list of the available parameters and their IDs.
Format:	SEP? [{<ItemID> <PamID>}]
Arguments:	<ItemID> is the element for querying a parameter value from nonvolatile memory. See below for details. <PamID> is the parameter ID, can be written in hexadecimal or decimal format. See below for details.
Response:	{<ItemID> <PamID>"="<PamValue> LF} where <PamValue> is the value of the specified parameter for the specified element
Troubleshooting:	Illegal element identifier, wrong parameter ID
Available element IDs and parameter IDs:	An element can be an axis, a channel or the entire system. The element type depends on the parameter. Refer to "Adapting Settings" (p. 257) for further information. Valid parameter IDs are specified in "Parameter Overview" (p. 267).

SPA (Set Volatile Memory Parameters)

Description:	Sets a parameter of the specified element in the volatile memory (RAM) to a specific value. Parameter changes are lost when the controller is switched off or rebooted.
Format:	SPA {<ItemID> <PamID> <PamValue>}
Arguments:	<ItemID> is the element for which a parameter is changed in volatile memory. See below for details. <PamID> is the parameter ID, can be written in hexadecimal or decimal format. See below for details. <PamValue> is the value to which the specified parameter of the specified element is set.
Response:	None
	Parameter changes are also lost when the parameters are reset to their default values with RPA (p. 195). **Note that this command is for setting hardware-specific parameters. Wrong values may lead to improper operation or damage of your hardware!** Related commands: HPA? (p. 184) returns a list of the available parameters. SEP (p. 197) writes parameter settings directly into nonvolatile memory (without changing the settings in volatile memory). WPA (p. 229) writes parameter settings from volatile to nonvolatile memory. RPA resets volatile memory to the value in nonvolatile memory.
Troubleshooting:	Illegal element identifier, wrong parameter ID, value out of range
Available element IDs and parameter IDs:	An element can be an axis, a channel or the entire system. The element type depends on the parameter. Refer to "Adapting Settings" (p. 257) for further information. Valid parameter IDs are specified in "Parameter Overview" (p. 267).
Example:	By default, control modes 6, 7, and 10 can be selected for the axes of the C-413 (see CMO (p. 159)). Further control modes can be made available for selection via the axis-related ***Available Closed-Loop Control Modes*** parameter (ID 0x07030101). For write access to this parameter and

the other ones mentioned in the following, it is necessary to switch to command level 1.

Send: `CCL 1 advanced`

Create a backup copy before changing parameter values, refer to "Creating and Loading a Backup Copy of Parameter Values" (p. 261).

Control mode 1 (direct PID position control) is to be activated for axis 2 via the ***Available Closed-Loop Control Modes*** parameter, and control mode 7 (PID position control with velocity control) is to be inhibited. The parameter value is bit-coded and must be set so that all permissible control modes are included (here 1, 6, and 10). The parameter ID and the value to be set are subsequently written in hexadecimal format.

Send: `SPA 2 0x07030101 0x00000442`

Note: Before control mode 1 is selected with CMO and servo mode is switched on with SVO, the servo control parameters for axis 2 must be adapted (***Position Servo P Term*** (ID 0x07000300), ***Position Servo I Term*** (ID 0x07000301) and ***Position Servo D Term*** (ID 0x07000302)). Otherwise, the axis will oscillate after the servo mode is switched on. When the control mode is changed from 7 to 1, the P term e.g. should first be reduced to one tenth of its original value, for example, the I term should be 0.01 and the D term should be 0.001. The servo control parameters are changed in the volatile memory with SPA. The following command can be necessary for the P term, for example:

Send: `SPA 2 0x07000300 10`

After the servo control parameters have been adapted and control mode 1 has been selected with CMO, the function of the system should be checked in closed-loop operation. If the performance of the closed-loop system proves to be satisfactory and you want to use the new system configuration as the default one, save the parameter settings from the volatile memory to the nonvolatile memory.

Send: `WPA 100`

Note: Refer to the command description for WPA (p. 229) for details on the extent of the saved settings.

SPA? (Get Volatile Memory Parameters)

Description: Queries the value of a parameter of a specified element from volatile memory (RAM).

You can obtain a list of the available parameters with HPA? (p. 184).

Format:	SPA? [{<ItemID> <PamID>}]
Arguments:	<ItemID> is the element for querying a parameter in volatile memory. See below for details. <PamID> is the parameter identifier, can be written in hexadecimal or decimal format. See below for details.
Response:	{<ItemID> <PamID>"="<PamValue> LF} where <PamValue> is the value of the specified parameter for the specified element
Troubleshooting:	Illegal element identifier, wrong parameter ID
Available elements IDs and parameter IDs:	An element can be an axis, a channel or the entire system. The element type depends on the parameter. Refer to "Adapting Settings" (p. 257) for further information. Valid parameter IDs are specified in "Parameter Overview" (p. 267).

SRG? (Query Status Register Value)

Description:	Returns register values for queried elements and registers.
Format:	SRG? [{<ItemID> <RegisterID>}]
Arguments:	<ItemID> is the element for querying a register. See below for details. <RegisterID> is the ID of the specified register; see below for available registers.
Response:	{<ItemID><RegisterID>"="<Value> LF} where <Value> is the value of the register; see below for more details.
Possible register IDs and response values:	Depending on the register bit, <ItemID> can be an axis or an input signal channel of the C-413; refer to the following table. <RegisterID> can be 1. <Value> is the bit-mapped response and is returned as the sum of the following individual codes in hexadecimal format:

Bit	15	14	13	12	11	10	9	8
Element*	a	a	a	a	-	i	i	i
Descrip-tion	On-target state	Determines the reference value	In motion	Servo mode on	-	Sensor signal valid	Reference edge found	Error Flag

Bit	7	6	5	4	3	2	1	0
Element*	-	-	-	-	i	-	i	-
Descrip-tion	-	-	-	-	Sensor is referenced	-	Reference switch	-

*a = axis, i = input signal channel

Example: Send: `SRG?`

Receive:

```
1 1=0x00000100
2 1=0x00009100
3 1=0x00000408
4 1=0x00000302
5 1=0x00000000
6 1=0x00000000
```

Note: The response is in hexadecimal format. This means: Axis 2 is on target (on-target status = true), and the servo mode for this axis is switched on. Input signal channel 3 (which is used to measure axis 2 in this example) is referenced, the sensor signal is valid, and the reference switch signal is low (i.e. axis 2 is located on the negative side of the reference switch). For input signal channel 4 (which is not used for measuring an axis here), the reference switch signal is high, and the reference edge has been found. The error flag is set for input signal channels 1, 2 and 4.

STE (Start Step And Response Measurement)

Description: Starts a step and records the step response for the specified axis.

The data recorder configuration, i.e., the assignment of data sources and record options to the recorder tables, can be set with DRC (p. 172).

The recorded data can be read with the DRR? command (p.

175).

Format: STE <AxisID> <Amplitude>

Arguments: <AxisID> is one axis of the controller

<Amplitude> is the size of the step. See below for details.

Response: None

Troubleshooting: The target value (closed-loop operation) or control value (open-loop operation) resulting from the specified step height is out of limits:

- Open-loop operation: The limitation of the control value results from the parameters 0x07000005 and 0x07000006.
- Closed-loop operation: Use CMN? (p. 159) and CMX? (p. 161) to get the currently valid limits.

Motion commands such as STE are not permitted when the analog control input or wave generator output is active for the axis. Refer to "Generating Control Values" (p. 24) for further information.

When the axis is in overflow state for more than 60 s (get with OVF? (p. 191)), the C-413 switches off the servo mode for the axis. Possible causes for the occurrence of the overflow state:

- When the control variable is the position or the velocity: The axis is blocked by an obstacle.
- When the control variable is the velocity or the force: The axis has reached the hard stop.

Notes: A "step" consists of a relative move of the specified amplitude.

How the value for <Amplitude> is interpreted depends on the current servo mode:

- Closed-loop operation: <Amplitude> is a relative target value in physical units. The control variable for which the relative target value is set with STE depends on the selected control mode (selection of the control mode, see CMO (p. 159)).
- Open-loop operation: <Amplitude> is a relative control value. The control value corresponds to the force in N to be generated.

Notes on protecting the mechanics connected:

- Closed-loop operation: When the control variable is the velocity or the force: The axis can move to the hard stop at a high velocity.

- Open-loop operation:
  - The axis can move to the hard stop at a high velocity.
  - When a high control value remains set over a long period of time, the mechanics can heat up. You can prevent the mechanics from overheating by activating I2t monitoring (p. 51).

STP (Stop All Axes)

Description:	Stops all axes abruptly. See the notes below for further details. Sets error code to 10. This command is identical in function to #24 (p. 150).
Format:	STP
Arguments:	None
Response:	None
Troubleshooting:	Communication breakdown
Notes:	#24 and STP stop all axis motion caused by motion commands (e.g. CTV (p. 169), CTR (p. 168), MOV (p. 188), VEL (p. 215), SVA (p. 205), SVR (p. 208)), commands for referencing (FRF (p. 178)), the wave generator (WGO (p. 224)), an analog control input and the autozero procedure (ATZ (p. 154)). The target value for the stopped axes in closed-loop operation is set as follows: ▪ The control variable is the position: The target value is set to the current value of the position. ▪ The control variable is the velocity or the force: The target value is set to zero. The control values for the stopped axes in open-loop operation are each set to the value of the ***autozero Result*** parameter (ID 0x07000A03, see ATZ (p. 154)). When the analog input is used as control source and the axis motion is stopped with STP (p. 204) or #24 (p. 150), the analog input channel is disconnected from the axis. To recommence commanding the axis via the analog input, the corresponding input signal channel must be reconnected to the axis. Refer to "Analog Input Signals" (p. 94) for further information.

SVA (Set Absolute Open-Loop Control Value)

Description:	Sets absolute open-loop control value to move the axis. Servo mode must be switched off when using this command (open-loop operation).
Format:	SVA {<AxisID> <ControlValueAbs>}
Arguments	<AxisID> is one axis of the controller. <ControlValueAbs> is the absolute control value for open-loop operation.
Response:	None
Troubleshooting:	The control value is out of limits. The limitation results from the parameters 0x07000005 and 0x07000006. Illegal axis identifier Servo mode is switched on for one of the specified axes Motion commands such as SVA are not permitted when the analog control input or wave generator output is active for the axis. Refer to "Generating Control Values" (p. 24) for further information.
Notes:	The control value for open-loop operation corresponds to the force in N to be generated. The motion can be stopped by #24 (p. 150), STP (p. 204) and HLT (p. 183). **Notes on protecting the mechanics connected:** ▪ The axis can move to the hard stop at a high velocity. When a high control value remains set over a long period of time, the mechanics can heat up. You can prevent the mechanics from overheating by activating I2t monitoring (p. 51).
Example 1:	Axis 1 is to generate a force of 2 N in open-loop operation. Send: SVA 1 2
Example 2:	Send: SVA 1 300 Send: ERR? Receive: 17 The axis does not move. The error code "17" indicates that the value for open-loop operation set by SVA is out of limits.

SVA? (Get Open-Loop Control Value)

Description: Queries the currently valid control value of the axis for open-loop operation.

Format: SVA? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<float> LF}

where

<float> is the currently valid control value for open-loop operation.

Troubleshooting: Illegal axis identifier

Notes: The control value corresponds to the force in N to be generated.

The control value for open-loop operation results from the following components:

- Immediately after the servo mode is switched off: Specification by the value of the ***AutoZero Result*** parameter (ID 0x07000A03, see ATZ (p. 154))
- Specification by control source, e.g. motion commands (SVA (p. 205), SVR (p. 208), IMP (p. 187), STE (p. 202)), analog control input or wave generator

Refer to "Generating Control Values" (p. 24) for further information.

SVO (Set Servo Mode)

Description: Sets the servo mode for specified axes (open-loop or closed-loop operation).

Format: SVO {<AxisID> <ServoState>}

Arguments: <AxisID> is one axis of the controller

<ServoState> can have the following values:
0 = servo mode off (open-loop operation)
1 = servo mode on (closed-loop operation)

Response: None

Troubleshooting: Illegal axis identifier

Notes: The variable that is controlled in closed-loop operation depends on the selected control mode (to select the control mode, see CMO (p. 159)).

When the servo mode is switched on, the target value for the control variable is set as follows to prevent jumps into the mechanics:

- The control variable is the position: The target value is set to the current value of the position.
- The control variable is the velocity or the force: The target value is set to zero.

When the servo mode is switched off, the control value of the axis is set to the value of the ***AutoZero Result*** parameter (ID 0x07000A03, see ATZ (p. 154)).

The current state of the servo mode determines the applicable motion commands:

- Servo mode on: CTV (p. 169), CTR (p. 168), IMP (p. 187) and STE (p. 202); depending on the control mode selected with CMO (p. 159) also MOV (p. 188), MVR (p. 189) or VEL (p. 215)
- Servo mode off: SVA (p. 205), SVR (p. 208), IMP (p. 187) and STE (p. 202)

In the following cases, it is not permitted to switch the servo mode on or off:

- The wave generator is running for the axis.
- An input signal channel is used as the control source for the axis.

The C-413 can be configured with the ***Power Up Servo Enable*** parameter (ID 0x07000800) so that the servo mode is automatically switched on upon switch-on or rebooting.

SVO? (Get Servo Mode)

Description:	Queries the servo mode for the axes specified. If arguments are not specified, queries the servo mode of all axes.
Format:	SVO? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response:	{<AxisID>"="<ServoState> LF} where <ServoState> is the current servo mode for the axis: 0 = servo mode off (open-loop operation) 1 = servo mode on (closed-loop operation)
Troubleshooting:	Illegal axis identifier

SVR (Set Relative Open-Loop Control Value)

Description:	Sets open-loop control value relative to the current open-loop control value to move the axis. Servo mode must be switched off when using this command (open-loop operation).
Format:	SVR {<AxisID> <ControlValueRel>}
Arguments	<AxisID> is one axis of the controller. <ControlValueRel> is the relative control value for open-loop operation. The sum of the relative control value and the currently valid control value for open-loop operation is set as the new absolute value for open-loop operation.
Response:	None
Troubleshooting:	The control value is out of limits. The limitation results from the parameters 0x07000005 and 0x07000006. Illegal axis identifier Servo mode is switched on for one of the specified axes Motion commands such as SVR are not permitted when the analog control input or wave generator output is active for the axis. Refer to "Generating Control Values" (p. 24) for further information.
Notes:	The control value for open-loop operation corresponds to the force in N to be generated. The motion can be stopped by #24 (p. 150), STP (p. 204) and HLT (p. 183). **Notes on protecting the mechanics connected:** ▪ The axis can move to the hard stop at a high velocity. ▪ When a high control value remains set over a long period of time, the mechanics can heat up. You can prevent the mechanics from overheating by activating I2t monitoring (p. 51).

TAD? (Get ADC Value Of Input Signal)

Description:	Get the current value from the specified input signal channel's A/D converter. Using this command it is possible to check for sensor overflow.
Format:	TAD? [{<InputSignalID>}]
Arguments:	<InputSignalID> is one input signal channel of the

	controller
Response:	{<InputSignalID>"="<uint> LF} where <uint> is the current A/D value, dimensionless
Notes:	The TAD? response represents the digitized signal value without filtering and linearization. Multiple input signal channels (sensors) can be involved in the measurement of one logical axis (see "Input matrix" (p. 16)).

TIO? (Tell Digital I/O Lines)

Description:	Tells number of installed digital I/O lines
Format:	TIO?
Arguments:	none
Response:	I=<uint1> O=<uint2> where <uint1> is the number of digital input lines. <uint2> is the number of digital output lines.
Notes:	All digital I/O lines are found on the **I/O** panel plug (p. 295) of the C-413. The digital output lines reported by TIO? are Output 1 to Output 5. The states of the lines can be set using the DIO command (p. 171). Furthermore, the lines can be programmed using the CTO command (p. 165) (trigger configuration) and the TRO command (p. 212) (trigger enabling/disabling). The Output 6 line of the **I/O** panel plug is not contained in the response to TIO?. This line outputs the servo cycles of the C-413 and is not accessible for commands. The digital input lines reported by TIO? are Input 1 to Input 4. The state of the lines can be queried with the DIO? command (p. 171). Furthermore, the lines can be programmed using the CTI command (p. 163) (trigger configuration) and the TRI command (p. 211) (trigger enabling/disabling).

TMN? (Get Minimum Commandable Position)

Description:	Get the minimum commandable position in physical units.
Format:	TMN? [{ <AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response	{<AxisID>"="<float> LF} where <float> is the minimum commandable position in physical units
Note:	The minimum commandable position is the value of the ***Position Range Limit min*** parameter (ID 0x07000000).

TMX? (Get Maximum Commandable Position)

Description:	Get the maximum commandable position in physical units.
Format:	TMX? [{ <AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response	{<AxisID>"="<float> LF} where <float> is the maximum commandable position in physical units
Note:	The maximum commandable position is the value of the ***Position Range Limit max*** parameter (ID 0x07000001).

TNR? (Get Number of Record Tables)

Description:	Queries the number of data recorder tables currently available on the controller.
Format:	TNR?
Arguments:	none
Response	<uint> is the number of data recorder tables which are currently available
Notes:	The response indicates the value of the ***Data Recorder Channel Number*** parameter (ID 0x16000300). The number of available data recorder tables can be set by changing the parameter value, refer to "Configuring the Data Recorder" (p. 83). Refer to "Data Recorder" (p. 83) for more information.

TNS? (Get Normalized Input Signal Value)

Description: Queries the normalized value for the specified input signal channel. This value functions internally as the input for mechanics linearization.

Several input signal channels (sensors) could be involved in the measurement of one logical axis (see "Input Matrix" (p. 16)).

Format: TNS? [{<InputSignalID>}]

Arguments: <InputSignalID> is one input signal channel of the controller

Response: {<InputSignalID>"="<float> LF}

where

<float> is the normalized, dimensionless value. The value range depends on the controller and the sensor type.

Notes: The C-413.20A and .2GA models have the additional input signal channels 5 and 6. These input signal channels can be used as sensor inputs or analog control inputs. The range of values for these channels is -100 to 100 for each and represents the voltage range from -10 to 10 V.

TPC? (Get Number of Output Signal Channels)

Description: Queries the number of output signal channels available in the controller.

Format: TPC?

Arguments: None

Response <uint> is the number of output signal channels available; the response specifies the value of the ***Number Of Output Channels*** parameter (ID 0x0E000B01).

Notes: The output signal channels consist of the drive channels and any additional analog output channels. The number of drive channels can be queried via the ***Number Of Driver Channels*** parameter (ID 0x0E000B04). Refer to "Commandable Elements" (p. 13) for further information.

TRI (Set Trigger Input State)

Description: Activates or deactivates the trigger configuration made with CTI (p. 163) for the specified digital input line.

Format: TRI {<TrigInID> <TrigInMode>}

Arguments: <TrigInID> is one digital input line of the controller; see below for further information.

<TrigInMode> can take on the following values:
0 = CTI trigger configuration deactivated
1 = CTI trigger configuration activated

Response:	None
Troubleshooting:	Illegal identifier of the digital input line
Notes:	<TrigInID> corresponds to the digital input lines Input 1 to Input 4, IDs = 1 to 4; refer to "I/O" (p. 295) for further information. The status of the digital input lines can be queried with DIO? (p. 171).

TRI? (Get Trigger Input State)

Description:	Queries the activation state of the trigger configuration made with CTI (p. 163) for the specified digital input line. If all arguments are left out, the state of all digital input lines is queried.
Format:	TRI? [{<TrigInID>}]
Arguments:	<TrigInID> is one digital input line of the controller; see the description of the TRI command (p. 211) for more information.
Response:	{<TrigInID>"="<TrigInMode> LF} where <TrigInMode> is the current state of the digital input line: 0 = CTI trigger configuration deactivated 1 = CTI trigger configuration activated
Troubleshooting:	Illegal identifier of the digital input line

TRO (Set Trigger Output State)

Description:	Activates or deactivates the trigger output conditions set with CTO (p. 165) for the specified digital output line.
Format:	TRO {<TrigOutID> <TrigMode>}
Arguments:	<TrigOutID> is a digital output line of the controller; see below for further details. <TrigMode> can have the following values: 0 = Trigger output deactivated 1 = Trigger output activated

Response:	None
Troubleshooting:	Illegal identifier of the digital output line
Notes:	<TrigOutID> corresponds to digital output lines Output 1 to Output 5, IDs = 1 to 5; refer to "I/O" (p. 295) for further information. Do not use DIO (p. 171) on digital output lines where the trigger output is activated with TRO.

TRO? (Get Trigger Output State)

Description:	Queries the activation status of the trigger output configuration made with CTO (p. 165) for the specified digital output line. If no arguments are specified, queries state of all digital output lines.
Format:	TRO? [{<TrigOutID>}]
Arguments:	<TrigOutID> is one digital output line of the controller, see TRO (p. 212) for more details.
Response:	{<TrigOutID>"="<TrigMode> LF} where <TrigMode> is the current state of the digital output line: 0 = Trigger output deactivated 1 = Trigger output activated
Troubleshooting:	Illegal identifier of the digital output line

TRS? (Indicate Reference Switch)

Description:	Indicates whether axes have a reference switch with direction sensing.
Format:	TRS? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	{<AxisID>"="<uint> LF} where <uint> indicates whether the axis has a direction-sensing reference switch (=1) or not (=0).
Troubleshooting:	Illegal axis identifier

TSC? (Get Number of Input Signal Channels)

Description:	Queries the number of input signal channels available in the controller.
Format:	TSC?
Arguments:	None
Response	<uint> is the number of input signal channels which are available; the response specifies the value of the ***Number Of Input Channels*** parameter (ID 0x0E000B00).
Notes:	The input signal channels consist of the sensor channels and any additional analog input channels. The number of sensor channels can be queried via the ***Number Of Sensor Channels*** parameter (ID 0x0E000B03). Refer to "Commandable Elements" (p. 13) for further information.

TSP? (Get Input Signal Position Value)

Description:	Queries the current position of the specified input signal channel. If no arguments are specified, the current position of all input signal channels is queried.
Format:	TSP? [{<InputSignalID>}]
Arguments:	<InputSignalID> is one input signal channel of the controller
Response:	{<InputSignalID>"="<float> LF} where <float> is the current position of the input signal channel, in physical units
Notes:	Multiple input signal channels (sensors) can be involved in the measurement of one logical axis (see "Input matrix" (p. 16)). Axis-related queries are possible with CAV? (p. 156) (current value of the control variable) and POS? (p. 193) (current position).

TWG? (Get Number of Wave Generators)

Description:	Queries the number of wave generators available in the controller.
Format:	TWG?
Arguments:	None
Response	<uint> is the number of wave generators which are

available

VEL (Set Closed-Loop Velocity)

Description: Set velocity of specified axes.

Format: VEL {<AxisID> <Velocity>}

Arguments: <AxisID> is one axis of the controller.

<Velocity> is the velocity value in physical units/s.

Response: None

Notes: With the C-413, velocity specifications are only effective in closed-loop operation.

The behavior of the axis when the velocity is set with VEL depends on the control variable (for selection of the control mode, see CMO (p. 159)):

- The control variable is the position or the force.
  - VEL can be used to set values from zero to the value of the ***Profile Generator Maximum Velocity*** parameter (ID 0x06010400). The velocity is set to the value of the parameter when the C-413 is switched on or rebooted and when you switch from velocity control to position or force control with CMO. Changing the parameter value in the volatile memory overwrites the velocity currently set with VEL.
  - The velocity can be changed with VEL while the axis is moving.
  - Setting the value zero with VEL stops the motion but does not change the current target value.
- The control variable is the velocity:
  - VEL specifies the target value of the velocity and triggers motion with the corresponding velocity. Target values with a positive and a negative sign are permissible. The sign of the target value determines the direction of motion. The value of the ***Profile Generator Maximum Velocity*** parameter is only used to limit the amount of the target value.
  - The target value of the velocity is set to zero when the C-413 is switched on or rebooted and when you switch from position or force control to velocity control with CMO.
  - The motion triggered by VEL can be stopped by setting the target value to 0 with VEL or by #24 (p. 150), STP (p. 204) and HLT (p. 183).

- VEL is not permissible when the analog control input or the wave generator output is active for the axis. Refer to "Generating Control Values" (p. 24) for further information.
- **Notes on protecting the connected mechanics:**
  The axis can move to the hard stop at high velocity. At the hard stop or when the axis is blocked by an obstacle, the overflow state can occur (get with OVF? (p. 191)). When the axis is in overflow state for more than 60 s, the C-413 switches the servo mode off for the axis.
  In the following case, the overflow state does **not** occur and the control value remains at maximum: The axis is at the hard stop or blocked, and the target value is set to zero or the motion is stopped. In this case, the servo mode for the axis must be switched off manually or motion in the opposite direction must be commanded, in order to reduce the control value and therefore prevent the mechanics from overheating.
  You can protect the mechanics from overheating vy activating I2t monitoring (p. 51).

Setting the velocity with VEL does not have any effect on the value of the ***Profile Generator Maximum Velocity*** parameter.

VEL? (Get Closed-Loop Velocity)

Description: Queries the commanded velocity.

If no arguments are specified, queries the value of all axes.

Format: VEL? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<float> LF}

where

<float> is the currently valid velocity value commanded in physical units per second.

Note: The velocity value queried with VEL? is only effective in closed-loop operation.

The interpretation of the velocity value queried with VEL? depends on the control variable (to select the control mode, see CMO (p. 159)):

- The control variable is the position or the force: VEL?

gets the currently valid maximum velocity.

- The control variable is the velocity: VEL? gets the currently valid target value of the velocity. The target value of the velocity can be changed by various sources, e.g,. by commands that cause motion (VEL (p. 215), CTV (p. 169), CTR (p. 168), STE (p. 202), IMP (p. 187)), by the wave generator or by an analog input signal. Refer to "Generating Control Values" (p. 24).

Further information, see VEL.

VOL? (Get Value Of Output Signal)

Description:	Queries the current value of the specified output signal channel in physical units.
Format:	VOL? [{<OutputSignalID>}]
Arguments:	<OutputSignalID> is one output signal channel of the controller
Response:	{<OutputSignalID>"="<float> LF} where <float> is the current value of the output signal channel in physical units.
Notes:	The queried value of output signal channels 1 and 2 corresponds to the output current in A. The C-413.20A and .2GA models have the additional output signal channels 3 and 4. These output signal channels can be used to monitor the position, force or velocity of an axis or to control an external motor driver; refer to "Analog Output Signals" (p. 110) for details. The queried value of output signal channels 3 and 4 corresponds to the output voltage in V. Multiple output signal channels (drive channels) can be involved in the motion of one logical axis (refer to "Allocating Axes to Channels" (p. 16)). Axis-related queries are possible with CCV? (p. 158) (current control value) for example.

WAV (Set Waveform Definition)

Description:	Defines a waveform of specified type for specified wave table.

To allow a flexible definition, a waveform (wave table contents) can be built up by stringing together "segments". Each segment is defined with a separate WAV command. A segment can be added to the existing wave table contents with the <AppendWave> argument (see below). To change individual segments or to modify their order, the complete waveform must be recreated segment-by-segment.

A segment can be based on predefined "curve" shapes (see the <WaveType> argument below).

Waveforms cannot be changed while they are being output by a wave generator. Before a waveform is modified with WAV, the wave generator output from the associated wave table must be stopped first.

The waveform values are absolute values.

The duration of one output cycle for the waveform can be calculated as follows:

Output Duration = Servo Cycle Time * WTR Value * Number of Points

where

Servo Cycle Time for the C-413 is specified by the parameter 0x0E000200 (in seconds)

WTR (wave table rate) value specifies the number of servo cycles the output of a waveform point lasts, default is 1

Number of Points corresponds to the wave table length (sum of the lengths of all segments in this table)

Refer to "Wave Generator" (p. 120) for more information.

Format: WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>

Arguments: <WaveTableID> is the wave table identifier.

<AppendWave> can be "X" or "&":
"X" clears the wave table and starts writing at the first point in the table.
"&" attaches the defined segment to the existing wave table contents in order to extend the waveform.

<WaveType> The type of curve used to define the segment. This can be one of
"PNT" (user-defined curve)

"SIN_P"(inverted cosine curve)
"RAMP" (ramp curve)
"LIN" (single scan line curve)

<WaveTypeParameters> stands for the parameters of the curve:

For "PNT":

<WaveStartPoint> <WaveLength> {<WavePoint>}

<WaveStartPoint>: The index of the starting point. Must be 1.

<WaveLength>: The number of points to be written in the wave table (= segment length).

<WavePoint>: The value of one single point.

For "SIN_P":

<SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <CurveCenterPoint>

<SegLength>: The length of the wave table segment in points. Only the number of points specified by <SegLength> will be written to the wave table. If the <SegLength> value is larger than the <WaveLength> value, the missing points in the segment are filled with the endpoint value of the curve.

<Amp>: The amplitude of the sine curve.

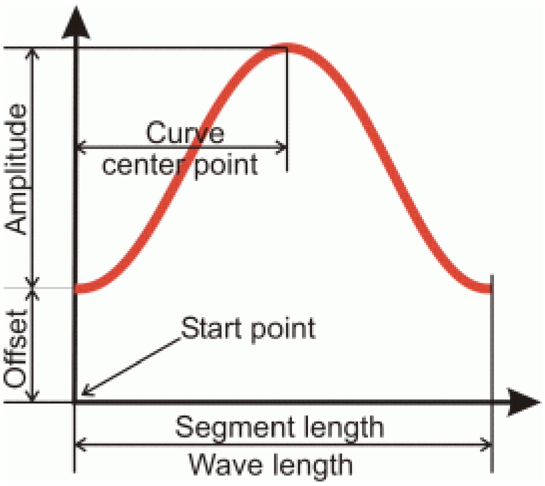
<Offset>: The offset of the sine curve.

<WaveLength>: The length of the sine curve in points.

<StartPoint>: The index of the starting point of the sine curve in the segment. Specifies the phase shift. Lowest possible value is 0.

<CurveCenterPoint>: The index of the center point of the sine curve. Determines if the curve is symmetrical or not. Lowest possible value is 0.

Example (refer to "Defining the Waveform" (p. 123) for further examples):



For "RAMP":

<SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <SpeedUpDown> <CurveCenterPoint>

<SegLength>: The length of the wave table segment in points. Only the number of points specified by <SegLength> will be written to the wave table. If the <SegLength> value is larger than the <WaveLength> value, the missing points in the segment are filled with the endpoint value of the curve.

<Amp>: The amplitude of the ramp curve.

<Offset>: The offset of the ramp curve.

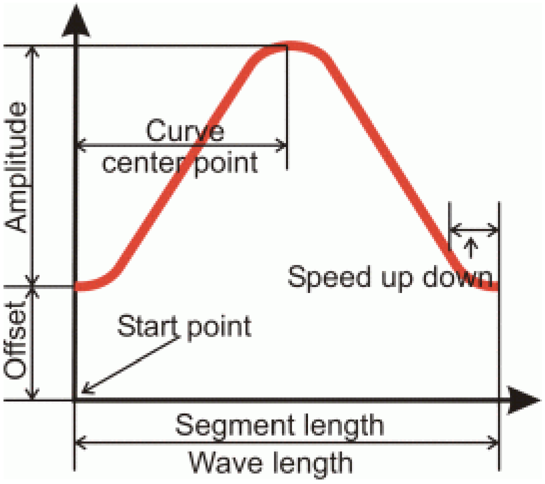
<WaveLength>: The length of the ramp curve in points.

<StartPoint>: The index of the starting point of the ramp curve in the segment. Specifies the phase shift. Lowest possible value is 0.

<SpeedUpDown>: The number of points for acceleration and delay.

<CurveCenterPoint>: The index of the center point of the ramp curve. Determines if the curve is symmetrical or not. Lowest possible value is 0.

Example (refer to "Defining the Waveform" (p. 123) for further examples):



For "LIN":

<SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <SpeedUpDown>

<SegLength>: The length of the wave table segment in points. Only the number of points specified by <SegLength> will be written to the wave table. If the <SegLength> value is larger than the <WaveLength> value, the missing points in the segment are filled with the endpoint value of the curve.

<Amp>: The amplitude of the scan line.

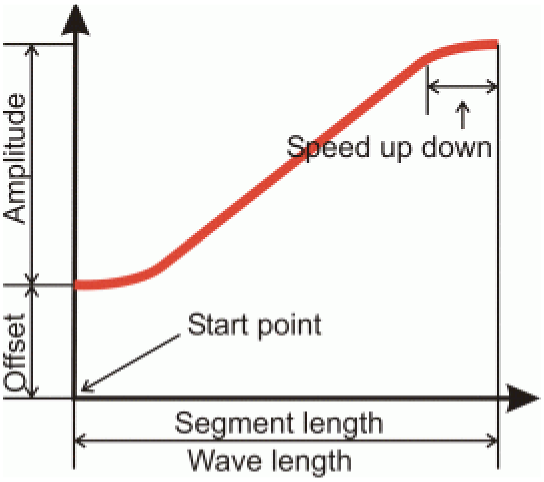
<Offset>: The offset of the scan line.

<WaveLength>: The length of the single scan line curve in points.

<StartPoint>: The index of the starting point of the scan line in the segment. Lowest possible value is 0.

<SpeedUpDown>: The number of points for acceleration and delay.

Example (refer to "Defining the Waveform" (p. 123) for further examples):



Note for the Sin_P, RAMP and LIN wave types:
If the <SegLength> value is larger than the <WaveLength> value, the missing points in the segment are filled with the endpoint value of the curve.

Response: None

Troubleshooting: Invalid wave table identifier

The total number of points for the waveform (which may consist of several segments) exceeds the available number of memory points.

Notes: The frequency of the wave generator output depends, among other factors, on the wave table length. The waveform must be selected so that the frequency of the wave generator output is smaller than the maximum permissible operating frequency of the connected mechanics (see specifications for the mechanics). When the frequency is too high, the motor driver in the C-413 can also overheat, and the output current is automatically switched off.

When defining a waveform with WAV, the resulting target values (closed-loop operation) or control values (open-loop operation) may exceed the respectively valid limit values:

- Open-loop operation: The limitation of the control value results from the parameters 0x07000005 and 0x07000006.
- Closed-loop operation: The currently valid limits can be queried with CMN? (p. 159) and CMX? (p. 161).

The amplitude is only limited during wave generator output: For points with a value that exceeds the respectively valid limit, the corresponding limit value is output. An error code is **not** set.

WAV? (Get Waveform Definition)

Description:	Queries the value of a wave parameter for a specified wave table. Refer to "Wave Generator" (p. 120) for more information.
Format:	WAV? [{<WaveTableID> <WaveParameterID>}]
Arguments:	<WaveTableID> is the wave table identifier. <WaveParameterID> is the wave parameter ID: 1 = Current wave table length as a number of points
Response:	{<WaveTableID> <WaveParameterID>"="<float> LF} where <float> depends on the <WaveParameterID>; specifies the current number of waveform points in the wave table for <WaveParameterID> = 1
Troubleshooting:	Invalid wave table identifier

WCL (Clear Wave Table Data)

Description:	Clears the content of the specified wave table. As long as a wave generator is running, it is not possible to clear the connected wave table. Refer to "Wave Generator" (p. 120) for more information.
Format:	WCL {<WaveTableID>}
Arguments:	<WaveTableID> is the wave table identifier.
Response:	None

WGC (Set Number Of Wave Generator Cycles)

Description:	Sets the number of output cycles for the specified wave generator (the output itself is started with WGO (p. 224)). Refer to "Wave Generator" (p. 120) for more information.
Format:	WGC {<WaveGenID> <Cycles>}
Arguments:	<WaveGenID> is the wave generator identifier

	<Cycles> is the number of wave generator output cycles.
Response:	None
Notes:	If cycles = 0 then the waveform is output without limitation until it is stopped by WGO or #24 (p. 150) or STP (p. 204). When the wave generator output is triggered by an external signal (WGO bit 1): The generator is stopped when the number of cycles specified by WGC is output. Further triggers are ignored.

WGC? (Get Number Of Wave Generator Cycles)

Description:	Queries the number of output cycles set for the specified wave generator. Refer to "Wave Generator" (p. 120) for more information.
Format:	WGC? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier
Response:	{<WaveGenID>"="<Cycles> LF} where <Cycles> is the number of wave generator output cycles set with WGC (p. 223).

WGO (Set Wave Generator Start/Stop Mode)

Description:	Starts and stops the specified wave generator in the specified mode. The number of output cycles can be limited by WGC (p. 223). You can lengthen the individual output cycles of the waveform with the WTR command (p. 231), . The wave generator output continues even after exiting the PC software that started it. The #9 command can be used to query the current activation state of the wave generators. WGO? queries the last start options commanded for the wave generator. Refer to "Wave Generator" (p. 120) for more information.

Format: WGO {<WaveGenID> <StartMode>}

Arguments: <WaveGenID> is the wave generator identifier

<StartMode> is the start mode for the specified wave generator.
In the WGO command, you supply the start mode in hex or decimal format. When no bits are set (<StartMode> = 0), there is no wave generator output for the associated axis. Note that bit 8 (0x100 or 256) cannot start the wave generator output by itself. It simply specifies a start option and must always be combined with one of the start modes specified in bit 0 (0x1 or 1) or bit 1 (0x2 or 2). See the examples below.

The start mode values in detail:

0: wave generator output is stopped. You can also stop the wave generator output with #24 (p. 150) or STP (p. 204).

bit 0 = 0x1 (hex format) or 1 (decimal format):
start wave generator output immediately, synchronized by servo cycle. In addition, one data recording cycle is started.

bit 1 = 0x2 (hex format) or 2 (decimal format):
start wave generator output triggered by external signal, synchronized by servo cycle.
The C-413's digital input lines can be used for providing the external signal (see the pin assignment for the I/O connector (p. 295)).
The trigger configuration is set with CTI (p. 163) and activated with TRI (p. 211).
Data recording can be started with WGR (p. 227) during wave generator output.

bit 8 = 0x100 (hex format) or 256 (decimal format):
wave generator started at the endpoint of the last cycle; start option.
The second and all subsequent output cycles each start at the endpoint of the preceding cycle which makes this start option appropriate to scanning applications.

Response: None

Troubleshooting: Invalid wave generator identifier

There is no wave table connected to the wave generator. Connect a wave table with WSL (p. 230).

Wave generator output and analog control input:
It is possible to configure an axis for control by an analog

input line while the wave generator output is active for that axis. In that case, the wave generator will continue to be active but its output will no longer be used for generating target or control values. As long as the corresponding axis is set up to be commanded by analog control input, the wave generator output can be stopped but not restarted.

Wave generator output and motion commands:
When the wave generator output is active, motion commands such as CTV (p. 169), MOV (p. 188) or SVA (p. 205) are not allowed for the associated axis.

Refer to "Generating Control Values" (p. 24) for further information.

Example: Wave generator 1 is to be used with the option "Start at the endpoint of the last cycle", i.e., bit 8 is switched on, whereby the value 0x100 (dec.: 256) is contributed to <StartMode>. Because bit 8 is only a "start option" and does not really start the wave generator output, a "start mode" ("immediately" or "triggered by external signal") must also be chosen. In this example, the wave generator is to be started by an external trigger signal, so bit 1 must be switched on, contributing 0x2 (dec.: 2), resulting in a <StartMode> value of 0x102 (dec.: 258).

Send the following WGO command, with the <StartMode> specified in hex format:

WGO 1 0x102

The same command with <StartMode> specified in decimal format:

WGO 1 258

To actually start the wave generator via an external trigger signal, it is also necessary to set and activate the trigger configuration with CTI (p. 163) and TRI (p. 211).

WGO? (Get Wave Generator Start/Stop Mode)

Description: Queries the start/stop mode of the specified wave generator.

Refer to "Wave Generator" (p. 120) for more information.

Format: WGO? [{<WaveGenID>}]

Arguments: <WaveGenID> is the wave generator identifier

Response: {<WaveGenID>"="<StartMode> LF}

where

<StartMode> is the last commanded start mode of the

	wave generator, in decimal format. Refer to WGO (p. 224) for further information.
Notes:	The value for <StartMode> can be the sum of several start options and one start mode, see WGO (p. 224). #24 (p. 150) and STP (p. 205) stop the wave generator output and set the start mode value to zero.

WGR (Starts Recording In Sync With Wave Generator)

Description:	Starts the data recording when the wave generator is active. Refer to "Wave Generator" (p. 120) and "Data Recorder" (p. 83) for more information.
Format:	WGR
Arguments:	None
Response:	None
Notes:	The data recorder can be configured with DRC (p. 172). The recorded data can be read with DRR? (p. 175) Starting the wave generator output with WGO (p. 224), bit 0, starts an initial data recording cycle at the same time. For further trigger options for starting the data recording, see DRT (p. 177).

WOS (Set Wave Generator Output Offset)

Description:	Sets an offset to the output of a wave generator. The current wave generator output is then created by adding the offset value to the current wave value: Generator Output = Offset + Current Wave Value Do not confuse the output-offset value set with WOS with the offset settings specified during waveform creation with WAV (p. 217). While the WAV offset affects only one segment (i.e., only one waveform), the WOS offset is added to all waveforms which are output by the specified wave generator. Deleting wave table content with WCL (p. 223) has no effect on the settings for the wave generator output offset. Refer to "Wave Generator" (p. 120) for more information.

Format:	WOS {<WaveGenID> <Offset>}
Arguments:	<WaveGenID> is the wave generator identifier <Offset> is the wave generator output offset, any float number. See below for details.
Response:	None
Notes:	WOS sets the value of the ***Wave Offset*** parameter (ID 0x1300010B) in the volatile memory. Required command level: 0. (Setting the parameter value with SPA (p. 199) requires switching to command level 1 with CCL (p. 157).) If the settings made with WOS are to be maintained when the C-413 is switched off or rebooted, they have to be saved with WPA (p. 229), refer also to "Adapting Settings" (p. 257). In closed-loop operation, the interpretation of the offset depends on the selected control mode (p. 28). In open-loop operation, the offset corresponds to the force in N to be generated, refer also to "Output matrix" (p. 19). When the resulting target value (closed-loop operation) or control value (open-loop operation) exceeds the respectively valid limit, the corresponding limit value is output. An error code is **not** set. ▪ Open-loop operation: The limitation results from the parameters 0x07000005 and 0x07000006. ▪ Closed-loop operation: The currently valid limits can be queried with CMN? (p. 159) and CMX? (p. 161).

WOS? (Get Wave Generator Output Offset)

Description:	Queries the current value of the offset which is added to the wave generator output. Refer to "Wave Generator" (p. 120) for more information.
Format:	WOS? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier
Response:	{<WaveGenID>"="<Offset> LF} where <Offset> is the current output offset of the wave generator. For interpretation of the value, see WOS (p. 227).
Notes:	The offset read by WOS? is the value of the ***Wave Offset***

parameter in the volatile memory (ID 0x1300010B).

WPA (Save Parameters To Non-Volatile Memory)

Description:	Writes the currently valid value of a parameter of a specified element from volatile memory (RAM) to nonvolatile memory. The values saved this way become the default values. **Note: If the current parameter values are incorrect, this can cause a fault in the system. Make sure that the parameter settings are correct before you execute the WPA command.** RAM settings not saved with WPA will be lost when the controller is switched off or rebooted or when RPA (p. 195) is used to restore the parameters. You can obtain a list of all available parameters with HPA? (p. 184). Use SPA? (p. 199) to check the current parameter settings in volatile memory. See SPA (p. 199) for an example.
Format:	WPA <Pswd> [{<ItemID> <PamID>}]
Arguments:	<Pswd> is the password for writing to the nonvolatile memory. See below for details. <ItemID> is the element for which a parameter is to be saved from the volatile to the nonvolatile memory. See below for details. <PamID> is the parameter identifier, can be written in hexadecimal or decimal format. See below for details.
Response:	None
Troubleshooting:	Illegal element identifier, wrong parameter ID, invalid password, command level too low for write access **Note that the number of write cycles in the nonvolatile memory is limited. Write default settings only if necessary.**
Notes:	Parameter values can be changed in the volatile memory with the SPA (p. 199), AOS (p. 151), ATZ (p. 154), CMO (p. 159), RTR (p. 195), WOS (p. 227) and WTR (p. 231) commands. Refer to "Adapting Settings" (p. 257) for further information.

When WPA is sent without specifying element and parameter IDs and only with the password, all currently valid parameter values are saved.

To have write access to the parameter(s), it may be necessary to switch to a higher command level using CCL (p. 157).

Note: Avoid switching the C-413 off during the WPA procedure.

Available passwords, element IDs and parameter IDs	The password for writing to the nonvolatile memory is "100". An element can be an axis, a channel or the entire system. The element type depends on the parameter. Refer to "Adapting Settings" (p. 257) for further information. Valid parameter IDs are specified in "Parameter Overview" (p. 267).

WSL (Set Connection Of Wave Table To Wave Generator)

Description:	Wave table selection: connects a wave table to a wave generator or disconnects the selected generator from any wave table. Two or more generators can be connected to the same wave table, but a generator cannot be connected to more than one wave table. Deleting wave table content with WCL (p. 223) has no effect on the WSL settings. As long as a wave generator is running, it is not possible to change its wave table connection. Refer to "Wave Generator" (p. 120) for more information.
Format:	WSL {<WaveGenID> <WaveTableID>}
Arguments:	<WaveGenID> is the wave generator identifier <WaveTableID> is the wave table identifier. If <WaveTableID> = 0, the selected generator is disconnected from any wave table.
Response:	None

WSL? (Get Connection Of Wave Table To Wave Generator)

Description:	Queries current wave table connection settings for the specified wave generator. Refer to "Wave Generator" (p. 120) for more information.
Format:	WSL? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier
Response:	{<WaveGenID>"="<WaveTableID> LF} where <WaveTableID> is the wave table identifier. If <WaveTableID> = 0, no wave table is connected to the wave generator.

WTR (Set Wave Generator Table Rate)

Description:	Sets wave generator table rate and interpolation type.
Format:	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}
Arguments:	<WaveGenID> is the wave generator identifier. See below for details. <WaveTableRate> is the wave generator table rate (unit: number of servo cycles); must be an integer value that is greater than zero <InterpolationType> Available interpolation types: See below.
Response:	None
Notes:	Different output rates can be set for the individual wave generators of the C-413. The output rate is set to the same value for all wave generators when <WaveGenID> has the value zero. WTR sets the value of the ***Wave Generator Table Rate*** parameter (ID 0x13000109) in the volatile memory. Required command level: 0. (Setting the parameter value with SPA (p. 199) requires switching to command level 1 with CCL (p. 157).) If the output rates set with WTR are to be preserved when the C-413 is switched off or rebooted, they have to be saved with WPA (p. 229); refer also to "Adapting Settings" (p. 257).

With the WTR command, the individual output cycles of the waveform can be lengthened. The duration of an output cycle for the waveform can be calculated as follows:

Output Duration = Servo Cycle Time * WTR Value * Number of Points

where

the servo cycle time for the C-413 is specified by the parameter 0x0E000200 (in seconds)

WTR value specifies the number of servo cycles the output of a waveform point lasts, default is 1

Number of Points is the length of the waveform (i.e. the length of the wave table)

The C-413 does not support any interpolation. <InterpolationType> must therefore be zero.

Refer to "Wave Generator" (p. 120) for more information. An application example can be found under "Configuring the Wave Generator" (p. 132).

WTR? (Get Wave Generator Table Rate)

Description: Queries the current wave generator table rate and the used interpolation type.

Refer to "Wave Generator" (p. 120) for more information. An application example can be found under "Configuring a Wave Generator" (p. 132).

Format: WTR? [{<WaveGenID>}]

Arguments: <WaveGenID> is the wave generator identifier

Response: {<WaveGenID>"="<WaveTableRate> <InterpolationType> LF}

where

<WaveTableRate> is the wave generator table rate (unit: Number of servo cycles)

<InterpolationType> is the interpolation type applied to outputs between wave table points when the output rate is higher than the minimum value. Refer to WTR (p. 231) for available interpolation types.

Notes:	The wave generator table rate read by WTR? is the ***Wave Generator Table Rate*** parameter value in the volatile memory (ID 0x13000109).

8.5 Error Codes

The error codes listed here are those of the PI General Command Set. As such, some may be not relevant to your controller and will simply never occur.

Controller Errors

0	PI_CNTR_NO_ERROR	No error
1	PI_CNTR_PARAM_SYNTAX	Parameter syntax error
2	PI_CNTR_UNKNOWN_COMMAND	Unknown command
3	PI_CNTR_COMMAND_TOO_LONG	Command length out of limits or command buffer overrun
4	PI_CNTR_SCAN_ERROR	Error while scanning
5	PI_CNTR_MOVE_WITHOUT_REF_OR_NO_SERVO	Unallowable move attempted on unreferenced axis, or move attempted with servo off
6	PI_CNTR_INVALID_SGA_PARAM	Parameter for SGA not valid
7	PI_CNTR_POS_OUT_OF_LIMITS	Position out of limits
8	PI_CNTR_VEL_OUT_OF_LIMITS	Velocity out of limits
9	PI_CNTR_SET_PIVOT_NOT_POSSIBLE	Attempt to set pivot point while U,V and W not all 0
10	PI_CNTR_STOP	Controller was stopped by command
11	PI_CNTR_SST_OR_SCAN_RANGE	Parameter for SST or for one of the embedded scan algorithms out of range
12	PI_CNTR_INVALID_SCAN_AXES	Invalid axis combination for fast scan
13	PI_CNTR_INVALID_NAV_PARAM	Parameter for NAV out of range
14	PI_CNTR_INVALID_ANALOG_INPUT	Invalid analog channel
15	PI_CNTR_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
16	PI_CNTR_INVALID_STAGE_NAME	Unknown stage name
17	PI_CNTR_PARAM_OUT_OF_RANGE	Parameter out of range
18	PI_CNTR_INVALID_MACRO_NAME	Invalid macro name
19	PI_CNTR_MACRO_RECORD	Error while recording macro

20	PI_CNTR_MACRO_NOT_FOUND	Macro not found
21	PI_CNTR_AXIS_HAS_NO_BRAKE	Axis has no brake
22	PI_CNTR_DOUBLE_AXIS	Axis identifier specified more than once
23	PI_CNTR_ILLEGAL_AXIS	Illegal axis
24	PI_CNTR_PARAM_NR	Incorrect number of parameters
25	PI_CNTR_INVALID_REAL_NR	Invalid floating point number
26	PI_CNTR_MISSING_PARAM	Parameter missing
27	PI_CNTR_SOFT_LIMIT_OUT_OF_RANGE	Soft limit out of range
28	PI_CNTR_NO_MANUAL_PAD	No manual pad found
29	PI_CNTR_NO_JUMP	No more step-response values
30	PI_CNTR_INVALID_JUMP	No step-response values recorded
31	PI_CNTR_AXIS_HAS_NO_REFERENCE	Axis has no reference sensor
32	PI_CNTR_STAGE_HAS_NO_LIM_SWITCH	Axis has no limit switch
33	PI_CNTR_NO_RELAY_CARD	No relay card installed
34	PI_CNTR_CMD_NOT_ALLOWED_FOR_STAGE	Command not allowed for selected stage(s)
35	PI_CNTR_NO_DIGITAL_INPUT	No digital input installed
36	PI_CNTR_NO_DIGITAL_OUTPUT	No digital output configured
37	PI_CNTR_NO_MCM	No more MCM responses
38	PI_CNTR_INVALID_MCM	No MCM values recorded
39	PI_CNTR_INVALID_CNTR_NUMBER	Controller number invalid
40	PI_CNTR_NO_JOYSTICK_CONNECTED	No joystick configured
41	PI_CNTR_INVALID_EGE_AXIS	Invalid axis for electronic gearing, axis can not be slave
42	PI_CNTR_SLAVE_POSITION_OUT_OF_RANGE	Position of slave axis is out of range
43	PI_CNTR_COMMAND_EGE_SLAVE	Slave axis cannot be commanded directly when electronic gearing is enabled
44	PI_CNTR_JOYSTICK_CALIBRATION_FAILED	Calibration of joystick failed
45	PI_CNTR_REFERENCING_FAILED	Referencing failed
46	PI_CNTR_OPM_MISSING	OPM (Optical Power Meter) missing
47	PI_CNTR_OPM_NOT_INITIALIZED	OPM (Optical Power Meter) not initialized or cannot be initialized
48	PI_CNTR_OPM_COM_ERROR	OPM (Optical Power Meter) Communication Error

49	PI_CNTR_MOVE_TO_LIMIT_SWITCH_FAILED	Move to limit switch failed
50	PI_CNTR_REF_WITH_REF_DISABLED	Attempt to reference axis with referencing disabled
51	PI_CNTR_AXIS_UNDER_JOYSTICK_CONTROL	Selected axis is controlled by joystick
52	PI_CNTR_COMMUNICATION_ERROR	Controller detected communication error
53	PI_CNTR_DYNAMIC_MOVE_IN_PROCESS	MOV! motion still in progress
54	PI_CNTR_UNKNOWN_PARAMETER	Unknown parameter
55	PI_CNTR_NO_REP_RECORDED	No commands were recorded with REP
56	PI_CNTR_INVALID_PASSWORD	Password invalid
57	PI_CNTR_INVALID_RECORDER_CHAN	Data Record Table does not exist
58	PI_CNTR_INVALID_RECORDER_SRC_OPT	Source does not exist; number too low or too high
59	PI_CNTR_INVALID_RECORDER_SRC_CHAN	Source Record Table number too low or too high
60	PI_CNTR_PARAM_PROTECTION	Protected Param: current Command Level (CCL) too low
61	PI_CNTR_AUTOZERO_RUNNING	Command execution not possible while Autozero is running
62	PI_CNTR_NO_LINEAR_AXIS	Autozero requires at least one linear axis
63	PI_CNTR_INIT_RUNNING	Initialization still in progress
64	PI_CNTR_READ_ONLY_PARAMETER	Parameter is read-only
65	PI_CNTR_PAM_NOT_FOUND	Parameter not found in non-volatile memory
66	PI_CNTR_VOL_OUT_OF_LIMITS	Voltage out of limits
67	PI_CNTR_WAVE_TOO_LARGE	Not enough memory available for requested wave curve
68	PI_CNTR_NOT_ENOUGH_DDL_MEMORY	Not enough memory available for DDL table; DDL can not be started
69	PI_CNTR_DDL_TIME_DELAY_TOO_LARGE	Time delay larger than DDL table; DDL can not be started
70	PI_CNTR_DIFFERENT_ARRAY_LENGTH	The requested arrays have different lengths; query them separately
71	PI_CNTR_GEN_SINGLE_MODE_RESTART	Attempt to restart the generator while it is running in single step mode

72	PI_CNTR_ANALOG_TARGET_ACTIVE	Motion commands and wave generator activation are not allowed when analog target is active
73	PI_CNTR_WAVE_GENERATOR_ACTIVE	Motion commands are not allowed when wave generator is active
74	PI_CNTR_AUTOZERO_DISABLED	No sensor channel or no piezo channel connected to selected axis (sensor and piezo matrix)
75	PI_CNTR_NO_WAVE_SELECTED	Generator started (WGO) without having selected a wave table (WSL).
76	PI_CNTR_IF_BUFFER_OVERRUN	Interface buffer did overrun and command couldn't be received correctly
77	PI_CNTR_NOT_ENOUGH_RECORDED_DATA	Data Record Table does not hold enough recorded data
78	PI_CNTR_TABLE_DEACTIVATED	Data Record Table is not configured for recording
79	PI_CNTR_OPENLOOP_VALUE_SET_WHEN_SERVO_ON	Open-loop commands (SVA, SVR) are not allowed when servo is on
80	PI_CNTR_RAM_ERROR	Hardware error affecting RAM
81	PI_CNTR_MACRO_UNKNOWN_COMMAND	Not macro command
82	PI_CNTR_MACRO_PC_ERROR	Macro counter out of range
83	PI_CNTR_JOYSTICK_ACTIVE	Joystick is active
84	PI_CNTR_MOTOR_IS_OFF	Motor is off
85	PI_CNTR_ONLY_IN_MACRO	Macro-only command
86	PI_CNTR_JOYSTICK_UNKNOWN_AXIS	Invalid joystick axis
87	PI_CNTR_JOYSTICK_UNKNOWN_ID	Joystick unknown
88	PI_CNTR_REF_MODE_IS_ON	Move without referenced stage
89	PI_CNTR_NOT_ALLOWED_IN_CURRENT_MOTION_MODE	Command not allowed in current motion mode
90	PI_CNTR_DIO_AND_TRACING_NOT_POSSIBLE	No tracing possible while digital IOs are used on this HW revision. Reconnect to switch operation mode.
91	PI_CNTR_COLLISION	Move not possible, would cause collision
92	PI_CNTR_SLAVE_NOT_FAST_ENOUGH	Stage is not capable of following the master. Check

		the gear ratio.
93	PI_CNTR_CMD_NOT_ALLOWED_WHILE_AXIS_IN_MOTION	This command is not allowed while the affected axis or its master is in motion.
94	PI_CNTR_OPEN_LOOP_JOYSTICK_ENABLED	Servo cannot be switched on when open-loop joystick control is activated.
95	PI_CNTR_INVALID_SERVO_STATE_FOR_PARAMETER	This parameter cannot be changed in current servo mode.
96	PI_CNTR_UNKNOWN_STAGE_NAME	Unknown stage name
97	PI_CNTR_INVALID_VALUE_LENGTH	Invalid length of value (too much characters)
98	PI_CNTR_AUTOZERO_FAILED	AutoZero procedure was not successful
99	PI_CNTR_SENSOR_VOLTAGE_OFF	Sensor voltage is off
100	PI_LABVIEW_ERROR	PI driver for use with NI LabVIEW reports error. See source control for details.
200	PI_CNTR_NO_AXIS	No stage connected to axis
201	PI_CNTR_NO_AXIS_PARAM_FILE	File with axis parameters not found
202	PI_CNTR_INVALID_AXIS_PARAM_FILE	Invalid axis parameter file
203	PI_CNTR_NO_AXIS_PARAM_BACKUP	Backup file with axis parameters not found
204	PI_CNTR_RESERVED_204	PI internal error code 204
205	PI_CNTR_SMO_WITH_SERVO_ON	SMO with servo on
206	PI_CNTR_UUDECODE_INCOMPLETE_HEADER	uudecode: incomplete header
207	PI_CNTR_UUDECODE_NOTHING_TO_DECODE	uudecode: nothing to decode
208	PI_CNTR_UUDECODE_ILLEGAL_FORMAT	uudecode: illegal UUE format
209	PI_CNTR_CRC32_ERROR	CRC32 error
210	PI_CNTR_ILLEGAL_FILENAME	Illegal file name (must be 8-0 format)
211	PI_CNTR_FILE_NOT_FOUND	File not found on controller
212	PI_CNTR_FILE_WRITE_ERROR	Error writing file on controller
213	PI_CNTR_DTR_HINDERS_VELOCITY_CHANGE	VEL command not allowed in DTR Command Mode
214	PI_CNTR_POSITION_UNKNOWN	Position calculations failed
215	PI_CNTR_CONN_POSSIBLY_BROKEN	The connection between controller and stage may be broken

216	PI_CNTR_ON_LIMIT_SWITCH	The connected stage has driven into a limit switch, some controllers need CLR to resume operation
217	PI_CNTR_UNEXPECTED_STRUT_STOP	Strut test command failed because of an unexpected strut stop
218	PI_CNTR_POSITION_BASED_ON_ESTIMATION	While MOV! is running position can only be estimated!
219	PI_CNTR_POSITION_BASED_ON_INTERPOLATION	Position was calculated during MOV motion
220	PI_CNTR_INTERPOLATION_FIFO_UNDERRUN	FIFO buffer underrun during interpolation
221	PI_CNTR_INTERPOLATION_FIFO_OVERFLOW	FIFO buffer overflow during interpolation
230	PI_CNTR_INVALID_HANDLE	Invalid handle
231	PI_CNTR_NO_BIOS_FOUND	No bios found
232	PI_CNTR_SAVE_SYS_CFG_FAILED	Save system configuration failed
233	PI_CNTR_LOAD_SYS_CFG_FAILED	Load system configuration failed
301	PI_CNTR_SEND_BUFFER_OVERFLOW	Send buffer overflow
302	PI_CNTR_VOLTAGE_OUT_OF_LIMITS	Voltage out of limits
303	PI_CNTR_OPEN_LOOP_MOTION_SET_WHEN_SERVO_ON	Open-loop motion attempted when servo ON
304	PI_CNTR_RECEIVING_BUFFER_OVERFLOW	Received command is too long
305	PI_CNTR_EEPROM_ERROR	Error while reading/writing EEPROM
306	PI_CNTR_I2C_ERROR	Error on I2C bus
307	PI_CNTR_RECEIVING_TIMEOUT	Timeout while receiving command
308	PI_CNTR_TIMEOUT	A lengthy operation has not finished in the expected time
309	PI_CNTR_MACRO_OUT_OF_SPACE	Insufficient space to store macro
310	PI_CNTR_EUI_OLDVERSION_CFGDATA	Configuration data has old version number
311	PI_CNTR_EUI_INVALID_CFGDATA	Invalid configuration data
333	PI_CNTR_HARDWARE_ERROR	Internal hardware error
400	PI_CNTR_WAV_INDEX_ERROR	Wave generator index error
401	PI_CNTR_WAV_NOT_DEFINED	Wave table not defined

402	PI_CNTR_WAV_TYPE_NOT_SUPPORTED	Wave type not supported
403	PI_CNTR_WAV_LENGTH_EXCEEDS_LIMIT	Wave length exceeds limit
404	PI_CNTR_WAV_PARAMETER_NR	Wave parameter number error
405	PI_CNTR_WAV_PARAMETER_OUT_OF_LIMIT	Wave parameter out of range
406	PI_CNTR_WGO_BIT_NOT_SUPPORTED	WGO command bit not supported
500	PI_CNTR_EMERGENCY_STOP_BUTTON_ACTIVATED	The \"red knob\" is still set and disables system
501	PI_CNTR_EMERGENCY_STOP_BUTTON_WAS_ACTIVATED	The \"red knob\" was activated and still disables system - reanimation required
502	PI_CNTR_REDUNDANCY_LIMIT_EXCEEDED	Position consistency check failed
503	PI_CNTR_COLLISION_SWITCH_ACTIVATED	Hardware collision sensor(s) are activated
504	PI_CNTR_FOLLOWING_ERROR	Strut following error occurred, e.g. caused by overload or encoder failure
505	PI_CNTR_SENSOR_SIGNAL_INVALID	One sensor signal is not valid
506	PI_CNTR_SERVO_LOOP_UNSTABLE	Servo loop was unstable due to wrong parameter setting and switched off to avoid damage.
507	PI_CNTR_LOST_SPI_SLAVE_CONNECTION	Digital connection to external SPI slave device is lost
508	PI_CNTR_MOVE_ATTEMPT_NOT_PERMITTED	Move attempt not permitted due to customer or limit settings
509	PI_CNTR_TRIGGER_EMERGENCY_STOP	Emergency stop caused by trigger input
530	PI_CNTR_NODE_DOES_NOT_EXIST	A command refers to a node that does not exist
531	PI_CNTR_PARENT_NODE_DOES_NOT_EXIST	A command refers to a node that has no parent node
532	PI_CNTR_NODE_IN_USE	Attempt to delete a node that is in use
533	PI_CNTR_NODE_DEFINITION_IS_CYCLIC	Definition of a node is cyclic
536	PI_CNTR_HEXAPOD_IN_MOTION	Transformation cannot be defined as long as Hexapod is in motion

537	PI_CNTR_TRANSFORMATION_TYPE_NOT_SUPPORTED	Transformation node cannot be activated
539	PI_CNTR_NODE_PARENT_IDENTICAL_TO_CHILD	A node cannot be linked to itself
540	PI_CNTR_NODE_DEFINITION_INCONSISTENT	Node definition is erroneous or not complete (replace or delete it)
542	PI_CNTR_NODES_NOT_IN_SAME_CHAIN	The nodes are not part of the same chain
543	PI_CNTR_NODE_MEMORY_FULL	Unused nodes must be deleted before new nodes can be stored
544	PI_CNTR_PIVOT_POINT_FEATURE_NOT_SUPPORTED	With some transformations pivot point usage is not supported
545	PI_CNTR_SOFTLIMITS_INVALID	Soft limits invalid due to changes in coordinate system
546	PI_CNTR_CS_WRITE_PROTECTED	Coordinate system is write protected
547	PI_CNTR_CS_CONTENT_FROM_CONFIG_FILE	Coordinate system cannot be changed because its content is loaded from a configuration file
548	PI_CNTR_CS_CANNOT_BE_LINKED	Coordinate system may not be linked
549	PI_CNTR_KSB_CS_ROTATION_ONLY	A KSB-type coordinate system can only be rotated by multiples of 90 degrees
551	PI_CNTR_CS_DATA_CANNOT_BE_QUERIED	This query is not supported for this coordinate system type
552	PI_CNTR_CS_COMBINATION_DOES_NOT_EXIST	This combination of work-and-tool coordinate systems does not exist
553	PI_CNTR_CS_COMBINATION_INVALID	The combination must consist of one work and one tool coordinate system
554	PI_CNTR_CS_TYPE_DOES_NOT_EXIST	This coordinate system type does not exist
555	PI_CNTR_UNKNOWN_ERROR	BasMac: unknown controller error
556	PI_CNTR_CS_TYPE_NOT_ACTIVATED	No coordinate system of this type is activated
557	PI_CNTR_CS_NAME_INVALID	Name of coordinate system is invalid

558	PI_CNTR_CS_GENERAL_FILE_MISSING	File with stored CS systems is missing or erroneous
559	PI_CNTR_CS_LEVELING_FILE_MISSING	File with leveling CS is missing or erroneous
601	PI_CNTR_NOT_ENOUGH_MEMORY	not enough memory
602	PI_CNTR_HW_VOLTAGE_ERROR	hardware voltage error
603	PI_CNTR_HW_TEMPERATURE_ERROR	hardware temperature out of range
604	PI_CNTR_POSITION_ERROR_TOO_HIGH	Position error of any axis in the system is too high
606	PI_CNTR_INPUT_OUT_OF_RANGE	Maximum value of input signal has been exceeded
607	PI_CNTR_NO_INTEGER	Value is not integer
608	PI_CNTR_FAST_ALIGNMENT_PROCESS_IS_NOT_RUNNING	Fast alignment process cannot be paused because it is not running
609	PI_CNTR_FAST_ALIGNMENT_PROCESS_IS_NOT_PAUSED	Fast alignment process cannot be restarted/resumed because it is not paused
650	PI_CNTR_UNABLE_TO_SET_PARAM_WITH_SPA	Parameter could not be set with SPA - SEP needed?
651	PI_CNTR_PHASE_FINDING_ERROR	Phase finding error
652	PI_CNTR_SENSOR_SETUP_ERROR	Sensor setup error
653	PI_CNTR_SENSOR_COMM_ERROR	Sensor communication error
654	PI_CNTR_MOTOR_AMPLIFIER_ERROR	Motor amplifier error
655	PI_CNTR_OVER_CURR_PROTEC_TRIGGERED_BY_I2T	Overcurrent protection triggered by I2T-module
656	PI_CNTR_OVER_CURR_PROTEC_TRIGGERED_BY_AMP_MODULE	Overcurrent protection triggered by amplifier module
657	PI_CNTR_SAFETY_STOP_TRIGGERED	Safety stop triggered
658	PI_SENSOR_OFF	Sensor off?
659	PI_CNTR_PARAM_CONFLICT	Parameter could not be set. Conflict with another parameter.
700	PI_CNTR_COMMAND_NOT_ALLOWED_IN_EXTERNAL_MODE	Command not allowed in external mode

710	PI_CNTR_EXTERNAL_MODE_ERROR	External mode communication error
715	PI_CNTR_INVALID_MODE_OF_OPERATION	Invalid mode of operation
716	PI_CNTR_FIRMWARE_STOPPED_BY_CMD	Firmware stopped by command (#27)
717	PI_CNTR_EXTERNAL_MODE_DRIVER_MISSING	External mode driver missing
718	PI_CNTR_CONFIGURATION_FAILURE_EXTERNAL_MODE	Missing or incorrect configuration of external mode
719	PI_CNTR_EXTERNAL_MODE_CYCLETIME_INVALID	External mode cycletime invalid
720	PI_CNTR_BRAKE_ACTIVATED	Brake is activated
725	PI_CNTR_DRIVE_STATE_TRANSITION_ERROR	Drive state transition error
731	PI_CNTR_SURFACEDETECTION_RUNNING	Command not allowed while surface detection is running
732	PI_CNTR_SURFACEDETECTION_FAILED	Last surface detection failed
733	PI_CNTR_FIELDBUS_IS_ACTIVE	Fieldbus is active and is blocking GCS control commands
1000	PI_CNTR_TOO_MANY_NESTED_MACROS	Too many nested macros
1001	PI_CNTR_MACRO_ALREADY_DEFINED	Macro already defined
1002	PI_CNTR_NO_MACRO_RECORDING	Macro recording not activated
1003	PI_CNTR_INVALID_MAC_PARAM	Invalid parameter for MAC
1004	PI_CNTR_RESERVED_1004	PI internal error code 1004
1005	PI_CNTR_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
1006	PI_CNTR_INVALID_IDENTIFIER	Invalid identifier (invalid special characters, ...)
1007	PI_CNTR_UNKNOWN_VARIABLE_OR_ARGUMENT	Variable or argument not defined
1008	PI_CNTR_RUNNING_MACRO	Controller is (already) running a macro

1009	PI_CNTR_MACRO_INVALID_OPERATOR	Invalid or missing operator for condition. Check necessary spaces around operator.
1010	PI_CNTR_MACRO_NO_ANSWER	No response was received while executing WAC/MEX/JRC/...
1011	PI_CMD_NOT_VALID_IN_MACRO_MODE	Command not valid during macro execution
1012	PI_CNTR_ERROR_IN_MACRO	Error occured during macro execution
1013	PI_CNTR_NO_MACRO_OR_EMPTY	No macro with given name on controller, or macro is empty
1015	PI_CNTR_INVALID_ARGUMENT	One or more arguments given to function is invalid (empty string, index out of range, ...)
1024	PI_CNTR_MOTION_ERROR	Motion error: position error too large, servo is switched off automatically
1025	PI_CNTR_MAX_MOTOR_OUTPUT_REACHED	Maximum motor output reached
1028	PI_CNTR_UNKNOWN_CHANNEL_IDENTIFIER	Unknown channel identifier
1063	PI_CNTR_EXT_PROFILE_UNALLOWED_CMD	User Profile Mode: Command is not allowed, check for required preparatory commands
1064	PI_CNTR_EXT_PROFILE_EXPECTING_MOTION_ERROR	User Profile Mode: First target position in User Profile is too far from current position
1065	PI_CNTR_PROFILE_ACTIVE	Controller is (already) in User Profile Mode
1066	PI_CNTR_PROFILE_INDEX_OUT_OF_RANGE	User Profile Mode: Block or Data Set index out of allowed range
1071	PI_CNTR_PROFILE_OUT_OF_MEMORY	User Profile Mode: Out of memory
1072	PI_CNTR_PROFILE_WRONG_CLUSTER	User Profile Mode: Cluster is not assigned to this axis
1073	PI_CNTR_PROFILE_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
1090	PI_CNTR_TOO_MANY_TCP_CONNECTIONS_OPEN	There are too many open tcpip connections

2000	PI_CNTR_ALREADY_HAS_SERIAL_NUMBER	Controller already has a serial number
2100	PI_CNTR_FEATURE_LICENSE_INVALID	Entered license is invalid
4000	PI_CNTR_SECTOR_ERASE_FAILED	Sector erase failed
4001	PI_CNTR_FLASH_PROGRAM_FAILED	Flash program failed
4002	PI_CNTR_FLASH_READ_FAILED	Flash read failed
4003	PI_CNTR_HW_MATCHCODE_ERROR	HW match code missing/invalid
4004	PI_CNTR_FW_MATCHCODE_ERROR	FW match code missing/invalid
4005	PI_CNTR_HW_VERSION_ERROR	HW version missing/invalid
4006	PI_CNTR_FW_VERSION_ERROR	FW version missing/invalid
4007	PI_CNTR_FW_UPDATE_ERROR	FW update failed
4008	PI_CNTR_FW_CRC_PAR_ERROR	FW Parameter CRC wrong
4009	PI_CNTR_FW_CRC_FW_ERROR	FW CRC wrong
5000	PI_CNTR_INVALID_PCC_SCAN_DATA	PicoCompensation scan data is not valid
5001	PI_CNTR_PCC_SCAN_RUNNING	PicoCompensation is running, some actions can not be executed during scanning/recording
5002	PI_CNTR_INVALID_PCC_AXIS	Given axis cannot be defined as PPC axis
5003	PI_CNTR_PCC_SCAN_OUT_OF_RANGE	Defined scan area is larger than the travel range
5004	PI_CNTR_PCC_TYPE_NOT_EXISTING	Given PicoCompensation type is not defined
5005	PI_CNTR_PCC_PAM_ERROR	PicoCompensation parameter error
5006	PI_CNTR_PCC_TABLE_ARRAY_TOO_LARGE	PicoCompensation table is larger than maximum table length
5100	PI_CNTR_NEXLINE_ERROR	Common error in NEXLINE® firmware module
5101	PI_CNTR_CHANNEL_ALREADY_USED	Output channel for NEXLINE® can not be redefined for other usage

5102	PI_CNTR_NEXLINE_TABLE_TOO_SMALL	Memory for NEXLINE® signals is too small
5103	PI_CNTR_RNP_WITH_SERVO_ON	RNP can not be executed if axis is in closed loop
5104	PI_CNTR_RNP_NEEDED	Relax procedure (RNP) needed
5200	PI_CNTR_AXIS_NOT_CONFIGURED	Axis must be configured for this action
5300	PI_CNTR_FREQU_ANALYSIS_FAILED	Frequency analysis failed
5301	PI_CNTR_FREQU_ANALYSIS_RUNNING	Another frequency analysis is running
6000	PI_CNTR_SENSOR_ABS_INVALID_VALUE	Invalid preset value of absolute sensor
6001	PI_CNTR_SENSOR_ABS_WRITE_ERROR	Error while writing to sensor
6002	PI_CNTR_SENSOR_ABS_READ_ERROR	Error while reading from sensor
6003	PI_CNTR_SENSOR_ABS_CRC_ERROR	Checksum error of absolute sensor
6004	PI_CNTR_SENSOR_ABS_ERROR	General error of absolute sensor
6005	PI_CNTR_SENSOR_ABS_OVERFLOW	Overflow of absolute sensor position

Interface Errors

0	COM_NO_ERROR	No error occurred during function call
-1	COM_ERROR	Error during com operation (could not be specified)
-2	SEND_ERROR	Error while sending data
-3	REC_ERROR	Error while receiving data
-4	NOT_CONNECTED_ERROR	Not connected (no port with given ID open)
-5	COM_BUFFER_OVERFLOW	Buffer overflow
-6	CONNECTION_FAILED	Error while opening port
-7	COM_TIMEOUT	Timeout error
-8	COM_MULTILINE_RESPONSE	There are more lines waiting in buffer
-9	COM_INVALID_ID	There is no interface or DLL handle with the given ID

-10	COM_NOTIFY_EVENT_ERROR	Event/message for notification could not be opened
-11	COM_NOT_IMPLEMENTED	Function not supported by this interface type
-12	COM_ECHO_ERROR	Error while sending "echoed" data
-13	COM_GPIB_EDVR	IEEE488: System error
-14	COM_GPIB_ECIC	IEEE488: Function requires GPIB board to be CIC
-15	COM_GPIB_ENOL	IEEE488: Write function detected no listeners
-16	COM_GPIB_EADR	IEEE488: Interface board not addressed correctly
-17	COM_GPIB_EARG	IEEE488: Invalid argument to function call
-18	COM_GPIB_ESAC	IEEE488: Function requires GPIB board to be SAC
-19	COM_GPIB_EABO	IEEE488: I/O operation aborted
-20	COM_GPIB_ENEB	IEEE488: Interface board not found
-21	COM_GPIB_EDMA	IEEE488: Error performing DMA
-22	COM_GPIB_EOIP	IEEE488: I/O operation started before previous operation completed
-23	COM_GPIB_ECAP	IEEE488: No capability for intended operation
-24	COM_GPIB_EFSO	IEEE488: File system operation error
-25	COM_GPIB_EBUS	IEEE488: Command error during device call
-26	COM_GPIB_ESTB	IEEE488: Serial poll-status byte lost
-27	COM_GPIB_ESRQ	IEEE488: SRQ remains asserted
-28	COM_GPIB_ETAB	IEEE488: Return buffer full
-29	COM_GPIB_ELCK	IEEE488: Address or board locked
-30	COM_RS_INVALID_DATA_BITS	RS-232: 5 data bits with 2 stop bits is an invalid combination, as is 6, 7, or 8 data bits with 1.5 stop bits

-31	COM_ERROR_RS_SETTINGS	RS-232: Error configuring the COM port
-32	COM_INTERNAL_RESOURCES_ERROR	Error dealing with internal system resources (events, threads, ...)
-33	COM_DLL_FUNC_ERROR	A DLL or one of the required functions could not be loaded
-34	COM_FTDIUSB_INVALID_HANDLE	FTDIUSB: invalid handle
-35	COM_FTDIUSB_DEVICE_NOT_FOUND	FTDIUSB: device not found
-36	COM_FTDIUSB_DEVICE_NOT_OPENED	FTDIUSB: device not opened
-37	COM_FTDIUSB_IO_ERROR	FTDIUSB: IO error
-38	COM_FTDIUSB_INSUFFICIENT_RESOURCES	FTDIUSB: insufficient resources
-39	COM_FTDIUSB_INVALID_PARAMETER	FTDIUSB: invalid parameter
-40	COM_FTDIUSB_INVALID_BAUD_RATE	FTDIUSB: invalid baud rate
-41	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_ERASE	FTDIUSB: device not opened for erase
-42	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_WRITE	FTDIUSB: device not opened for write
-43	COM_FTDIUSB_FAILED_TO_WRITE_DEVICE	FTDIUSB: failed to write device
-44	COM_FTDIUSB_EEPROM_READ_FAILED	FTDIUSB: EEPROM read failed
-45	COM_FTDIUSB_EEPROM_WRITE_FAILED	FTDIUSB: EEPROM write failed
-46	COM_FTDIUSB_EEPROM_ERASE_FAILED	FTDIUSB: EEPROM erase failed
-47	COM_FTDIUSB_EEPROM_NOT_PRESENT	FTDIUSB: EEPROM not present
-48	COM_FTDIUSB_EEPROM_NOT_PROGRAMMED	FTDIUSB: EEPROM not programmed
-49	COM_FTDIUSB_INVALID_ARGS	FTDIUSB: invalid arguments
-50	COM_FTDIUSB_NOT_SUPPORTED	FTDIUSB: not supported
-51	COM_FTDIUSB_OTHER_ERROR	FTDIUSB: other error
-52	COM_PORT_ALREADY_OPEN	Error while opening the COM port: was already open
-53	COM_PORT_CHECKSUM_ERROR	Checksum error in received data from COM port
-54	COM_SOCKET_NOT_READY	Socket not ready, you should call the function again
-55	COM_SOCKET_PORT_IN_USE	Port is used by another socket
-56	COM_SOCKET_NOT_CONNECTED	Socket not connected (or not valid)
-57	COM_SOCKET_TERMINATED	Connection terminated (by

		peer)
-58	COM_SOCKET_NO_RESPONSE	Can't connect to peer
-59	COM_SOCKET_INTERRUPTED	Operation was interrupted by a nonblocked signal
-60	COM_PCI_INVALID_ID	No device with this ID is present
-61	COM_PCI_ACCESS_DENIED	Driver could not be opened (on Vista: run as administrator!)
-62	COM_SOCKET_HOST_NOT_FOUND	Host not found
-63	COM_DEVICE_CONNECTED	Device already connected
-64	COM_INVALID_COM_PORT	Invalid COM port
-65	COM_USB_DEVICE_NOT_FOUND	USB device not found
-66	COM_NO_USB_DRIVER	No USB driver installed
-67	COM_USB_NOT_SUPPORTED	USB is not supported

DLL Errors

-1001	PI_UNKNOWN_AXIS_IDENTIFIER	Unknown axis identifier
-1002	PI_NR_NAV_OUT_OF_RANGE	Number for NAV out of range--must be in [1,10000]
-1003	PI_INVALID_SGA	Invalid value for SGA--must be one of 1, 10, 100, 1000
-1004	PI_UNEXPECTED_RESPONSE	Controller sent unexpected response
-1005	PI_NO_MANUAL_PAD	No manual control pad installed, calls to SMA and related commands are not allowed
-1006	PI_INVALID_MANUAL_PAD_KNOB	Invalid number for manual control pad knob
-1007	PI_INVALID_MANUAL_PAD_AXIS	Axis not currently controlled by a manual control pad
-1008	PI_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g., reference move, fast scan algorithm)
-1009	PI_THREAD_ERROR	Internal error--could not start thread
-1010	PI_IN_MACRO_MODE	Controller is (already) in macro mode--command not valid in macro mode
-1011	PI_NOT_IN_MACRO_MODE	Controller not in macro

		mode--command not valid unless macro mode active
-1012	PI_MACRO_FILE_ERROR	Could not open file to write or read macro
-1013	PI_NO_MACRO_OR_EMPTY	No macro with given name on controller, or macro is empty
-1014	PI_MACRO_EDITOR_ERROR	Internal error in macro editor
-1015	PI_INVALID_ARGUMENT	One or more arguments given to function is invalid (empty string, index out of range, ...)
-1016	PI_AXIS_ALREADY_EXISTS	Axis identifier is already in use by a connected stage
-1017	PI_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
-1018	PI_COM_ARRAY_ERROR	Could not access array data in COM server
-1019	PI_COM_ARRAY_RANGE_ERROR	Range of array does not fit the number of parameters
-1020	PI_INVALID_SPA_CMD_ID	Invalid parameter ID given to SPA or SPA?
-1021	PI_NR_AVG_OUT_OF_RANGE	Number for AVG out of range--must be >0
-1022	PI_WAV_SAMPLES_OUT_OF_RANGE	Incorrect number of samples given to WAV
-1023	PI_WAV_FAILED	Generation of wave failed
-1024	PI_MOTION_ERROR	Motion error: position error too large, servo is switched off automatically
-1025	PI_RUNNING_MACRO	Controller is (already) running a macro
-1026	PI_PZT_CONFIG_FAILED	Configuration of PZT stage or amplifier failed
-1027	PI_PZT_CONFIG_INVALID_PARAMS	Current settings are not valid for desired configuration
-1028	PI_UNKNOWN_CHANNEL_IDENTIFIER	Unknown channel identifier
-1029	PI_WAVE_PARAM_FILE_ERROR	Error while reading/writing wave generator parameter file
-1030	PI_UNKNOWN_WAVE_SET	Could not find description of wave form. Maybe WG.INI is missing?
-1031	PI_WAVE_EDITOR_FUNC_NOT_LOADED	The WGWaveEditor DLL function was not found at startup
-1032	PI_USER_CANCELLED	The user cancelled a dialog

-1033	PI_C844_ERROR	Error from C-844 Controller
-1034	PI_DLL_NOT_LOADED	DLL necessary to call function not loaded, or function not found in DLL
-1035	PI_PARAMETER_FILE_PROTECTED	The open parameter file is protected and cannot be edited
-1036	PI_NO_PARAMETER_FILE_OPENED	There is no parameter file open
-1037	PI_STAGE_DOES_NOT_EXIST	Selected stage does not exist
-1038	PI_PARAMETER_FILE_ALREADY_OPENED	There is already a parameter file open. Close it before opening a new file
-1039	PI_PARAMETER_FILE_OPEN_ERROR	Could not open parameter file
-1040	PI_INVALID_CONTROLLER_VERSION	The version of the connected controller is invalid
-1041	PI_PARAM_SET_ERROR	Parameter could not be set with SPA--parameter not defined for this controller!
-1042	PI_NUMBER_OF_POSSIBLE_WAVES_EXCEEDED	The maximum number of wave definitions has been exceeded
-1043	PI_NUMBER_OF_POSSIBLE_GENERATORS_EXCEEDED	The maximum number of wave generators has been exceeded
-1044	PI_NO_WAVE_FOR_AXIS_DEFINED	No wave defined for specified axis
-1045	PI_CANT_STOP_OR_START_WAV	Wave output to axis already stopped/started
-1046	PI_REFERENCE_ERROR	Not all axes could be referenced
-1047	PI_REQUIRED_WAVE_NOT_FOUND	Could not find parameter set required by frequency relation
-1048	PI_INVALID_SPP_CMD_ID	Command ID given to SPP or SPP? is not valid
-1049	PI_STAGE_NAME_ISNT_UNIQUE	A stage name given to CST is not unique
-1050	PI_FILE_TRANSFER_BEGIN_MISSING	A uuencoded file transferred did not start with "begin" followed by the proper filename
-1051	PI_FILE_TRANSFER_ERROR_TEMP_FILE	Could not create/read file on host PC
-1052	PI_FILE_TRANSFER_CRC_ERROR	Checksum error when

		transferring a file to/from the controller
-1053	PI_COULDNT_FIND_PISTAGES_DAT	The PiStages.dat database could not be found. This file is required to connect a stage with the CST command
-1054	PI_NO_WAVE_RUNNING	No wave being output to specified axis
-1055	PI_INVALID_PASSWORD	Invalid password
-1056	PI_OPM_COM_ERROR	Error during communication with OPM (Optical Power Meter), maybe no OPM connected
-1057	PI_WAVE_EDITOR_WRONG_PARAMNUM	WaveEditor: Error during wave creation, incorrect number of parameters
-1058	PI_WAVE_EDITOR_FREQUENCY_OUT_OF_RANGE	WaveEditor: Frequency out of range
-1059	PI_WAVE_EDITOR_WRONG_IP_VALUE	WaveEditor: Error during wave creation, incorrect index for integer parameter
-1060	PI_WAVE_EDITOR_WRONG_DP_VALUE	WaveEditor: Error during wave creation, incorrect index for floating point parameter
-1061	PI_WAVE_EDITOR_WRONG_ITEM_VALUE	WaveEditor: Error during wave creation, could not calculate value
-1062	PI_WAVE_EDITOR_MISSING_GRAPH_COMPONENT	WaveEditor: Graph display component not installed
-1063	PI_EXT_PROFILE_UNALLOWED_CMD	User Profile Mode: Command is not allowed, check for required preparatory commands
-1064	PI_EXT_PROFILE_EXPECTING_MOTION_ERROR	User Profile Mode: First target position in User Profile is too far from current position
-1065	PI_EXT_PROFILE_ACTIVE	Controller is (already) in User Profile Mode
-1066	PI_EXT_PROFILE_INDEX_OUT_OF_RANGE	User Profile Mode: Block or Data Set index out of allowed range
-1067	PI_PROFILE_GENERATOR_NO_PROFILE	ProfileGenerator: No profile has been created yet
-1068	PI_PROFILE_GENERATOR_OUT_OF_LIMITS	ProfileGenerator: Generated profile exceeds limits of one

		or both axes
-1069	PI_PROFILE_GENERATOR_UNKNOWN_PARAMETER	ProfileGenerator: Unknown parameter ID in Set/Get Parameter command
-1070	PI_PROFILE_GENERATOR_PAR_OUT_OF_RANGE	ProfileGenerator: Parameter out of allowed range
-1071	PI_EXT_PROFILE_OUT_OF_MEMORY	User Profile Mode: Out of memory
-1072	PI_EXT_PROFILE_WRONG_CLUSTER	User Profile Mode: Cluster is not assigned to this axis
-1073	PI_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
-1074	PI_INVALID_DEVICE_DRIVER_VERSION	The installed device driver doesn't match the required version. Please see the documentation to determine the required device driver version.
-1075	PI_INVALID_LIBRARY_VERSION	The library used doesn't match the required version. Please see the documentation to determine the required library version.
-1076	PI_INTERFACE_LOCKED	The interface is currently locked by another function. Please try again later.
-1077	PI_PARAM_DAT_FILE_INVALID_VERSION	Version of parameter DAT file does not match the required version. Current files are available at www.pi.ws.
-1078	PI_CANNOT_WRITE_TO_PARAM_DAT_FILE	Cannot write to parameter DAT file to store user defined stage type.
-1079	PI_CANNOT_CREATE_PARAM_DAT_FILE	Cannot create parameter DAT file to store user defined stage type.
-1080	PI_PARAM_DAT_FILE_INVALID_REVISION	Parameter DAT file does not have correct revision.
-1081	PI_USERSTAGES_DAT_FILE_INVALID_REVISION	User stages DAT file does not have correct revision.
-1082	PI_SOFTWARE_TIMEOUT	Timeout Error. Some lengthy operation did not finish within expected time.
-1083	PI_WRONG_DATA_TYPE	A function argument has an unexpected data type.
-1084	PI_DIFFERENT_ARRAY_SIZES	Length of data arrays is different.

-1085	PI_PARAM_NOT_FOUND_IN_PARAM_DAT_FILE	Parameter value not found in parameter DAT file.
-1086	PI_MACRO_RECORDING_NOT_ALLOWED_IN_THIS_MODE	Macro recording is not allowed in this mode of operation.
-1087	PI_USER_CANCELLED_COMMAND	Command cancelled by user input.
-1088	PI_TOO_FEW_GCS_DATA	Controller sent too few GCS data sets
-1089	PI_TOO_MANY_GCS_DATA	Controller sent too many GCS data sets
-1090	PI_GCS_DATA_READ_ERROR	Communication error while reading GCS data
-1091	PI_WRONG_NUMBER_OF_INPUT_ARGUMENTS	Wrong number of input arguments.
-1092	PI_FAILED_TO_CHANGE_CCL_LEVEL	Change of command level has failed.
-1093	PI_FAILED_TO_SWITCH_OFF_SERVO	Switching off the servo mode has failed.
-1094	PI_FAILED_TO_SET_SINGLE_PARAMETER_WHILE_PERFORMING_CST	A parameter could not be set while performing CST: CST was not performed (parameters remain unchanged).
-1095	PI_ERROR_CONTROLLER_REBOOT	Connection could not be reestablished after reboot.
-1096	PI_ERROR_AT_QHPA	Sending HPA? or receiving the response has failed.
-1097	PI_QHPA_NONCOMPLIANT_WITH_GCS	HPA? response does not comply with GCS2 syntax.
-1098	PI_FAILED_TO_READ_QSPA	Response to SPA? could not be received.
-1099	PI_PAM_FILE_WRONG_VERSION	Version of PAM file cannot be handled (too old or too new)
-1100	PI_PAM_FILE_INVALID_FORMAT	PAM file does not contain required data in PAM-file format
-1101	PI_INCOMPLETE_INFORMATION	Information does not contain all required data
-1102	PI_NO_VALUE_AVAILABLE	No value for parameter available
-1103	PI_NO_PAM_FILE_OPEN	No PAM file is open
-1104	PI_INVALID_VALUE	Invalid value
-1105	PI_UNKNOWN_PARAMETER	Unknown parameter

-1106	PI_RESPONSE_TO_QSEP_FAILED	Response to SEP? could not be received.
-1107	PI_RESPONSE_TO_QSPA_FAILED	Response to SPA? could not be received.
-1108	PI_ERROR_IN_CST_VALIDATION	Error while performing CST: One or more parameters were not set correctly.
-1109	PI_ERROR_PAM_FILE_HAS_DUPLICATE_ENTRY_WITH_DIFFERENT_VALUES	PAM file has duplicate entry with different values.
-1110	PI_ERROR_FILE_NO_SIGNATURE	File has no signature
-1111	PI_ERROR_FILE_INVALID_SIGNATURE	File has invalid signature
-1112	PI_ERROR_CANNOT_DETERMINE_ACTUAL_END_OF_TRAVEL_WHILE_PLATFORM_IS_MOVING	Cannot determine actual end of travel range while platform is moving.
-1113	PI_ERROR_AT_QIDN	Sending IDN? or receiving the response has failed.
-1114	PI_ERROR_AT_MAC_DEF	Sending MAC_DEF or receiving the response has failed.
-1115	PI_CONTROLLER_OR_CONTROLLER_VERSION_DOES_NOT_EXIST_IN_PISTAGES_DATABASE	Sending Controller or controller version does not exist in PIStages database.
-1116	PI_NOT_ENOUGH_MEMORY	Not enough memory
-1117	PI_ERROR_AXIS_RUNTIME_ERROR	Runtime error indicated for axis, check error log with \"LOG?\" to find more details.
-1118	PI_ERROR_SYSTEM_RUNTIME_CRITICAL_ERROR	Critical error indicated for system, check error log with \"LOG?\" to find more details.
-1119	PI_ERROR_CANNOT_START_EMULATOR	Cannot start emulation software.
-1120	COM_DEVICE_NOT_SUPPORTED	Device is not supported
-10000	PI_PARAMETER_DB_INVALID_STAGE_TYPE_FORMAT	PI stage database: String containing stage type and description has invalid format.
-10001	PI_PARAMETER_DB_SYSTEM_NOT_AVAILABLE	PI stage database: Database does not contain the selected stage type for the connected controller.
-10002	PI_PARAMETER_DB_FAILED_TO_ESTABLISH_CONNECTION	PI stage database: Establishing the connection has failed.

-10003	PI_PARAMETER_DB_COMMUNICATION_ERROR	PI stage database: Communication was interrupted (e.g. because database was deleted).
-10004	PI_PARAMETER_DB_ERROR_WHILE_QUERYING_PARAMETERS	PI stage database: Querying data failed.
-10005	PI_PARAMETER_DB_SYSTEM_ALREADY_EXISTS	PI stage database: System already exists. Rename stage and try again.
-10006	PI_PARAMETER_DB_QHPA_CONTANS_UNKNOWN_PAM_IDS	PI stage database: Response to HPA? contains unknown parameter IDs.
-10007	PI_PARAMETER_DB_AND_QHPA_ARE_INCONSISTENT	PI stage database: Inconsistency between database and response to HPA?.
-10008	PI_PARAMETER_DB_SYSTEM_COULD_NOT_BE_ADDED	PI stage database: Stage has not been added.
-10009	PI_PARAMETER_DB_SYSTEM_COULD_NOT_BE_REMOVED	PI stage database: Stage has not been removed.
-10010	PI_PARAMETER_DB_CONTROLLER_DB_PARAMETERS_MISMATCH	Controller does not support all stage parameters stored in PI stage database. No parameters were set.
-10011	PI_PARAMETER_DB_DATABASE_IS_OUTDATED	The version of PISTAGES3.DB stage database is out of date. Please update via PIUpdateFinder. No parameters were set.
-10012	PI_PARAMETER_DB_AND_HPA_MISMATCH_STRICT	Mismatch between number of parameters present in stage database and available in controller interface. No parameters were set.
-10013	PI_PARAMETER_DB_AND_HPA_MISMATCH_LOOSE	Mismatch between number of parameters present in stage database and available in controller interface. Some parameters were ignored.
-10014	PI_PARAMETER_DB_FAILED_TO_SET_PARAMETERS_CORRECTLY	One or more parameters could not be set correctly on the controller.

-10015	PI_PARAMETER_DB_MISSING_PARAMETER_DEFINITIONS_IN_DATABASE	One or more parameter definitions are not present in stage database. Please update PISTAGES3.DB via PIUpdateFinder. Missing parameters were ignored.
-10016	PI_PARAMETER_DB_MISSING_FIRMWARE_FEATURE_ON_CONTROLLER	Parameters could not be set on controller because the corresponding firmware feature is missing

9 Adapting Settings

The properties of the C-413 and the connected mechanics are stored in the C-413 as parameter values (e.g., settings for matrices (p. 16), selected control mode (p. 28), settings for the servo algorithm (p. 33)).

The parameters can be divided into the following categories:

- Protected parameters with default settings that cannot be changed
- Parameters that can be set by the user to adapt to the application

Write permission for the parameters is determined by command levels (p. 157).

Each parameter is in both the C-413's volatile and nonvolatile memory. The values in the nonvolatile memory are loaded to the volatile memory as default values when switching on or rebooting the C-413. The values in the volatile memory determine the current behavior of the system.

9.1 Changing Parameter Values in the C-413

NOTICE

Unsuitable parameter settings!

The values in the nonvolatile memory are loaded to the volatile memory as default values when switching on or rebooting the C-413 and take effect immediately. Unsuitable parameter settings can cause damage to the connected mechanics.

- ➢ Change parameter values only after careful consideration.
- ➢ Save the current parameter values to the PC (p. 261) before you make changes in the nonvolatile memory.

INFORMATION

The number of write cycles in the nonvolatile memory is restricted by the limited lifetime of the memory chip (EEPROM).

- ➢ Overwrite the default values only when it is necessary.
- ➢ Save the current parameter values to the PC (p. 261) before you make changes in the nonvolatile memory.
- ➢ Contact our customer service department (p. 287), if the C-413 exhibits unexpected behavior.

INFORMATION

If the mechanics have an ID chip (p. 53), the data is loaded from the ID chip into the volatile and nonvolatile memory of the C-413 after switching on or rebooting the C-413.

Parameters that are loaded from the ID chip are marked in color in the parameter overview (p. 267).

9.1.1 Commands for Parameters

The commands for getting, modifying and saving parameters can be divided into the following:

- Commands that apply to all parameters
- Commands for fast access to individual parameters

Commands for all parameters

Command	Syntax	Function
`CCL`	CCL <Level> [<PSWD>]	Change to a higher command level, e.g. to obtain write permission for particular parameters.
`CCL?`	CCL?	Gets the active command level.
`HPA?`	HPA?	Responds with a help string which contains all available parameters with short descriptions.
`HPV?`	HPV?	Responds with a help string which contains possible parameters values.
`RPA`	RPA [{<ItemID> <PamID>}]	Copy a parameter value from the nonvolatile to the volatile memory.
`SEP`	SEP <Pswd> {<ItemID> <PamID> <PamValue>}	Modify a parameter value in the nonvolatile memory.
`SEP?`	SEP? [{<ItemID> <PamID>}]	Get parameter values from the nonvolatile memory.
`SPA`	SPA {<ItemID> <PamID> <PamValue>}	Modify a parameter value in the volatile memory.
`SPA?`	SPA? [{<ItemID> <PamID>}]	Get parameter values from the volatile memory.
`WPA`	WPA <Pswd> [{<ItemID> <PamID>}]	Copy a current parameter value from the volatile to the nonvolatile memory. Here it is used as a default value.

Commands for fast access to individual parameters

Command	Syntax	Function
`AOS`	AOS {<AxisID> <Offset>}	Sets the value of the ***Analog Target Offset*** parameter (ID 0x06000501) in the volatile memory. Required command level: 0. (Setting the parameter value with SPA (p. 199) requires switching to command level 1 with CCL (p. 157).) Refer to "Analog Input Signals" (p. 94) for further information.
`AOS?`	AOS? [{<AxisID>}]	Gets the value of the ***Analog Target Offset*** parameter (ID 0x06000501) in the volatile memory. Refer to "Analog Input Signals" (p. 94) for further information.

Command	Syntax	Function
`ATZ`	ATZ [{<AxisID> <LowValue>}]	The adjustment procedure started with ATZ changes the value of the ***AutoZero Result*** parameter (ID 0x07000A03) in the volatile memory. When a force sensor is allocated to the axis via the ***Input Channel For Force Feedback*** parameter (ID 0x07000400), the adjustment procedure also changes the value of the ***Sensor Mech. Correction 1*** parameter (ID 0x02000200) for the input signal channel of the force sensor. The parameter value is changed in the volatile memory. Refer to "Autozero Procedure for Compensating Weight Force" (p. 49) for further information.
`CMN?`	CMN? [{<AxisID>}]	Gets the following parameters in the volatile memory depending on the selected control mode: ▪ Control variable is the position: Value of the ***Position Range Limit min*** parameter (ID 0x07000000) ▪ Control variable is the velocity: Value of the ***Profile Generator Maximum Velocity*** parameter (ID 0x06010400) with negative sign ▪ Control variable is the force: Value of the ***Force Range Limit min*** parameter (ID 0x07000005) Refer to "Generating Control Values" (p. 24) for further information.
`CMO`	CMO {<AxisID> <CtrlMode>}	Sets the value of the ***Closed-Loop Control Mode*** parameter (ID 0x07030100) in the volatile memory. Required command level: 0. (Setting the parameter value with SPA (p. 199) requires switching to command level 1 with CCL (p. 157).) Refer to "Control Modes and Control Variables" (p. 28) for more information.
`CMO?`	CMO? [{<AxisID>}]	Gets the value of the ***Closed-Loop Control Mode*** parameter (ID 0x07030100) in the volatile memory. Refer to "Control Modes and Control Variables" (p. 28) for more information.
`CMX?`	CMX? [{<AxisID>}]	Gets the following parameters in the volatile memory depending on the selected control mode: ▪ Control variable is the position: Value of the ***Position Range Limit max*** parameter (ID 0x07000001) ▪ Control variable is the velocity: Value of the ***Profile Generator Maximum Velocity*** parameter (ID 0x06010400) with positive sign ▪ Control variable is the force: Value of the ***Force Range Limit max*** parameter (ID 0x07000006)

Command	Syntax	Function
		Refer to "Generating Control Values" (p. 24) for further information.
`RTR`	RTR <RecordTableRate>	Sets the value of the ***Data Recorder Table Rate*** parameter (ID 0x16000000) in the volatile memory. Refer to "Data Recorder" (p. 83) for more information.
`RTR?`	RTR?	Gets the value of the ***Data Recorder Table Rate*** parameter (ID 0x16000000) in the volatile memory. Refer to "Data Recorder" (p. 83) for more information.
`TMN?`	TMN? [{<AxisID>}]	Gets the value of the ***Position Range Limit min*** parameter (ID 0x07000000) in the volatile memory. Refer to "Generating Control Values" (p. 24) for further information.
`TMX?`	TMX? [{<AxisID>}]	Gets the value of the ***Position Range Limit max*** parameter (ID 0x07000001) in the volatile memory. Refer to "Generating Control Values" (p. 24) for further information.
`TNR?`	TNR?	Gets the value of the ***Data Recorder Channel Number*** parameter (ID 0x16000300) in the volatile memory. Refer to "Data Recorder" (p. 83) for further information.
`TPC?`	TPC?	Gets the value of the ***Number Of Output Channels*** parameter (ID 0x0E000B01). Refer to "Commandable Elements" (p. 13) for further information.
`TSC?`	TSC?	Gets the value of the ***Number Of Input Channels*** parameter (ID 0x0E000B00). Refer to "Commandable Elements" (p. 13) for further information.
`WOS`	WOS {<WaveGenID> <Offset>}	Sets the value of the ***Wave Offset*** parameter (ID 0x1300010B) in the volatile memory. Required command level: 0. (Setting the parameter value with SPA (p. 199) requires switching to command level 1 with CCL (p. 157).) Refer to "Wave Generator" (p. 120) for further information.
`WOS?`	WOS? [{<WaveGenID>}]	Gets the value of the ***Wave Offset*** parameter (ID 0x1300010B) in the volatile memory. Refer to "Wave Generator" (p. 120) for further information.
`WTR`	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}	Sets the value of the ***Wave Generator Table Rate*** parameter (ID 0x13000109) in the volatile memory. Required command level: 0. (Setting the parameter value with SPA (p. 199) requires switching to

Command	Syntax	Function
		command level 1 with CCL (p. 157).) Refer to "Wave Generator" (p. 120) for further information.
`WTR?`	WTR? [{<WaveGenID>}]	Gets the value of the ***Wave Generator Table Rate*** parameter (ID 0x13000109) in the volatile memory. Refer to "Wave Generator" (p. 120) for further information.

You can find details in the command descriptions (p. 148).

9.1.2 Creating and Loading a Backup Copy of Parameter Values

The PC software from PI allows you to back up the C-413's parameter values to your PC and load the saved values back to the C-413. Options:

- PIMikroMove provides the following options in the ***Device Parameter Configuration*** window:
  - Saving parameter values in parameter files (file extension .pam) (p. 262)
  - Loading parameter values from parameter files (file extension .pam) (p. 262)
- PITerminal allows parameter values to be saved in text files (file extension .txt) (p. 261). It is **not** possible to load parameter values directly from text files.

INFORMATION

The properties of the C-413 and the connected mechanics are stored in the C-413 as parameter values (e.g., settings for matrices (p. 16), selected control mode (p. 28), settings for the servo algorithm (p. 33)).

➢ Create a backup copy on the PC before changing the parameter values of the C-413. You can then restore the original settings at any time.

➢ Create an additional backup copy with a new file each time after optimizing the parameter values.

Requirements

✓ You have read and understood the General Notes on Startup (p. 69).

✓ You have established communication between the C-413 and the PC with PIMikroMove or PITerminal (p. 72).

✓ When you create the backup in the ***Device Parameter Configuration*** window of PIMikroMove:

  – You have read and understood "Device Parameter Configuration" in the PIMikroMove manual.

Saving parameter values in a parameter file with PIMikroMove

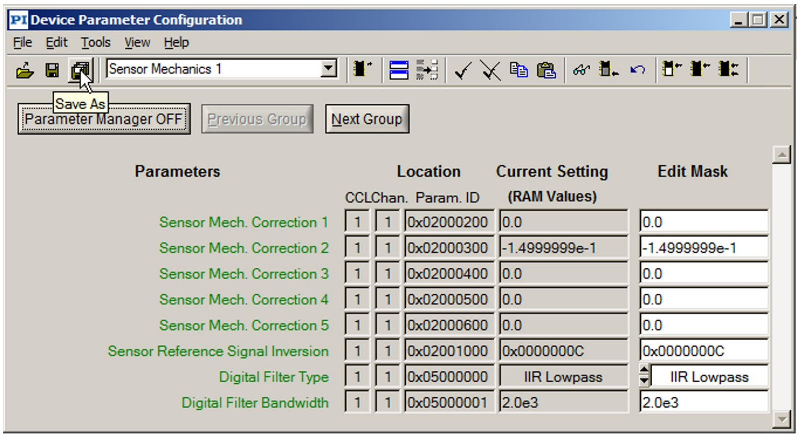


Figure 37: Example: Device Parameter Configuration window in PIMikroMove, mouse pointer is on the Save As button

1. Open the ***Device Parameter Configuration*** window in PIMikroMove:
   – In the main window, select the ***C-413 > Parameter Configuration…*** menu item.
2. Make sure that the ***Edit Mask*** column in the ***Device Parameter Configuration*** window has the correct parameter values.
3. Save the parameter values from the ***Edit Mask*** column of the ***Device Parameter Configuration*** window in a parameter file (file extension .pam) on your PC. Use one of the following options:
   – ***File > Save Edit Values*** or ***File > Save Edit Values As*** menu item
   – (Save) or (Save As) button in the icon bar

Loading parameter values from a parameter file with PIMikroMove

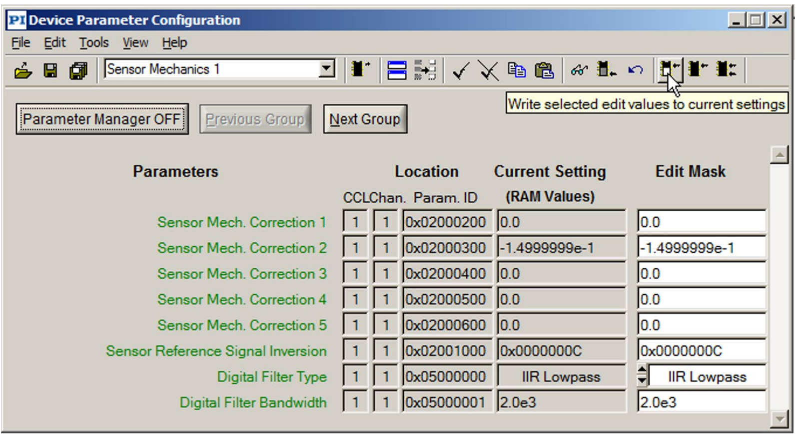


Figure 38: Example: Device Parameter Configuration window in PIMikroMove, mouse pointer is on the button for loading to the volatile memory

1. Open the ***Device Parameter Configuration*** window in PIMikroMove:

– In the main window, select the ***C-413 > Parameter Configuration…*** menu item.

2. Load the parameter values from a parameter file on the PC to the ***Edit Mask*** column of the ***Device Parameter Configuration*** window:

   a) Open the file selection window via the ***File > Load and select*** menu item of the ***Device Parameter Configuration*** window.

   b) In the file selection window, select the parameter file (file extension .pam) to be loaded on the PC.

   c) Start the loading in the file selection window with the ***Load*** button.

3. Make sure that the ***Edit Mask*** column in the ***Device Parameter Configuration*** window has correct parameter values.

4. In the ***Device Parameter Configuration*** window, select the parameter values that are to be loaded to the C-413 from the ***Edit Mask*** column.

   If you want to load all parameter values from the parameter file (also comprising the parameter groups that are not currently displayed):

   – Select the parameter values via the ***Edit > Select all > for all Parameters*** menu item.

   If you only want to load the parameter values from the parameter group that is currently displayed in the ***Device Parameter Configuration*** window, use one of the following options:

   – Select individual fields in the ***Edit Mask*** column.

   – Select all fields in the ***Edit Mask*** column: button in the icon bar or the ***Edit > Select all > for this Parameter Type*** menu item.

   The selected parameter values have a gray background in the ***Edit Mask*** column.

5. Load the selected parameter values to the corresponding memory of the C-413:

   – To the volatile memory: button in the icon bar or the ***Tools > Write selected Edit Values > to Current Setting (RAM)*** menu item

   – To the nonvolatile memory: button in the icon bar or the ***Tools > Write selected Edit Values > to Default Setting (EEPROM)*** menu item

   – To the volatile and nonvolatile memory: button in the icon bar or the ***Tools > Write selected Edit Values > to Default and Current Setting (EEPROM + RAM)*** menu item

   When the ***Password*** dialog opens:

   – Enter the password *advanced* to change to command level 1.

Saving parameter values in a text file with PITerminal

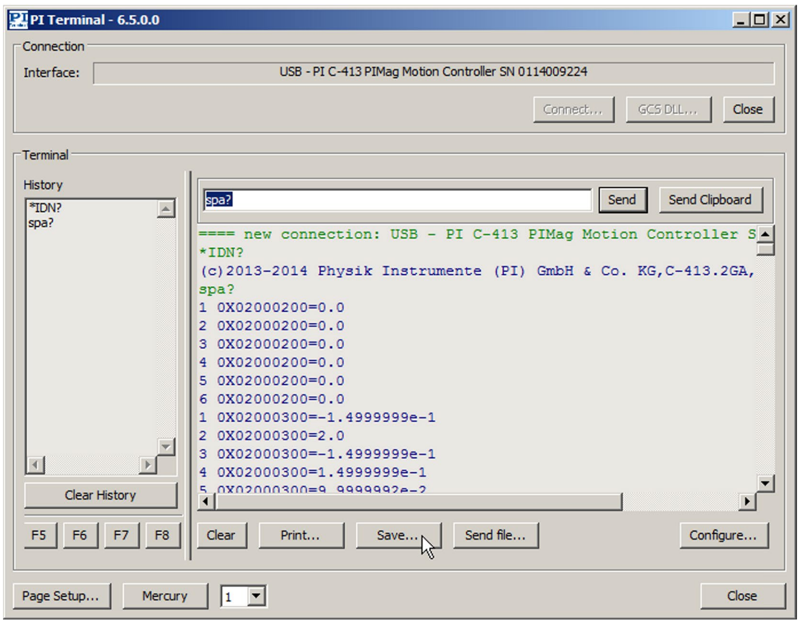


Figure 39: Example: Main window of PITerminal, the parameter values have been queried in the volatile memory with SPA?, mouse pointer is on the Save... button

In PITerminal the main window from which commands can be sent is opened automatically after establishing communication.

1. Get the parameter values from which you want to create a backup copy.
   - If you want to save the parameter values from the volatile memory of the C-413: Send the `SPA?` command.
   - If you want to save the parameter values from the nonvolatile memory of the C-413: Send the `SEP?` command.
2. Select ***Save...***.

   The ***Save content of terminal as textfile*** window opens.
3. In the ***Save content of terminal as textfile*** window, save the queried parameter values in a text file on your PC.

9.1.3 Changing Parameter Values: General Procedure

NOTICE

Unsuitable parameter settings!

The values in the nonvolatile memory are loaded to the volatile memory as default values when switching on or rebooting the C-413 and take effect immediately. Unsuitable parameter settings can cause damage to the connected mechanics.

- Change parameter values only after careful consideration.
- Save the current parameter values to the PC (p. 261) before you make changes in the nonvolatile memory.

INFORMATION

Write access for the parameters of the C-413 is defined by command levels. After the controller is switched on or rebooted, the active command level is always level 0. For particular parameters, write access is only allowed on command level 1. On command levels > 1, write access is only available to PI service personnel.

The C-413 ignores the active command level in the following cases:

- The C-413 reads the parameter values from the ID chip of the mechanics.
- The settings of the C-413 are changed with the AOS (p. 151), CMO (p. 159), WOS (p. 227) or WTR (p. 231) commands.
- The current parameter values are written from the volatile to the nonvolatile memory (directly with WPA or in the PC software).

- If necessary, send the `CCL 1 advanced` command or enter the password `advanced` to change to command level 1.
- Contact the customer service department if there seem to be problems with parameters of command level 2 or higher (p. 287).

INFORMATION

To change parameter values, you can use commands or work in the ***Device Parameter Configuration*** window of PIMikroMove.

- Get an overview of the available parameters in "Parameter Overview" (p. 267).

If you want to use commands:

- Get an overview of the available commands in "Commands for Parameters" (p. 258).
- Follow the instructions in:
  - "Getting available parameters" (p. 266)
  - "Getting and modifying parameter values in the nonvolatile memory" (p. 266)
  - "Getting and modifying parameter values in the volatile memory" (p. 266)
  - "Writing parameter values from the volatile memory to the nonvolatile memory" (p. 266)
  - "Writing parameter values from the nonvolatile memory to the volatile memory" (p. 267)

If you want to work in the ***Device Parameter Configuration*** window of PIMikroMove:

- Read "Device Parameter Configuration" in the PIMikroMove manual.
- Determine, modify and save parameter values with the corresponding buttons and menu items in the ***Device Parameter Configuration*** window of PIMikroMove.

Getting available parameters

The parameters available for adapting the C-413 to your application depend on the firmware of your C-413.

- Send the `HPA?` command (p. 184) to obtain a list of all available parameters with a short description.

  Among other things, this list specifies which command level allows write access to the parameter value.

- Send the `HPV?` command (p. 185) to obtain a list of possible parameter values.

  The response to `HPV?` is empty when all required information is already contained in the response to `HPA?`.

Getting and modifying parameter values in the nonvolatile memory

- Send the `SEP?` command (p. 198) to obtain a list of parameter values in the nonvolatile memory.
- To modify the parameter values in the nonvolatile memory, proceed as follows:

  a) Create a backup copy of the current parameter values (p. 261).

  b) Check the new parameter values carefully.

  c) Modify the parameter values in the nonvolatile memory with the `SEP` command (p. 197) using the password *100*.

Getting and modifying parameter values in the volatile memory

- Send the `SPA?` command (p. 200) to obtain a list of parameter values in the volatile memory.
- Modify the parameter values in the volatile memory with the following commands:
  - Usable for all parameters: The `SPA` command (p. 199).
  - Only usable for selected parameters: Commands for fast access to parameters; see corresponding table in "Commands for Parameters" (p. 258).

Writing parameter values from the volatile memory to the nonvolatile memory

1. Create a backup copy of the current parameter values (p. 261).
2. Modify the parameter values in the volatile memory; refer to "Getting and modifying parameter values in the volatile memory" (p. 266).
3. Check whether the C-413 functions correctly with the modified parameter values.

   If so:

   - Write the modified parameter values to the nonvolatile memory with the `WPA` command (p. 229) using the password *100*.

If not:

– Change and check the parameters in the volatile memory again.

Writing parameter values from the nonvolatile memory to the volatile memory

1. Send the `SEP?` command (p. 198) to obtain a list of parameter values in the nonvolatile memory.
2. Check the parameter values in the response to the `SEP?` command.
3. If you are sure that the C-413 will function correctly with the parameter values from the nonvolatile memory:
   – Write the parameter values from the nonvolatile memory to the volatile memory with the `RPA` command (p. 195).

9.2 Parameter Overview

INFORMATION

Write access for the parameters of the C-413 is defined by command levels. After the controller is switched on or rebooted, the active command level is always level 0. For particular parameters, write access is only allowed on command level 1. On command levels > 1, write access is only available to PI service personnel.

The C-413 ignores the active command level in the following cases:

- The C-413 reads the parameter values from the ID chip of the mechanics.
- The settings of the C-413 are changed with the AOS (p. 151), CMO (p. 159), WOS (p. 227) or WTR (p. 231) commands.
- The current parameter values are written from the volatile to the nonvolatile memory (directly with WPA or in the PC software).

➢ If necessary, send the `CCL 1 advanced` command or enter the password `advanced` to change to command level 1.

➢ Contact the customer service department if there seem to be problems with parameters of command level 2 or higher (p. 287).

INFORMATION

To save parameter values in the nonvolatile memory, it is necessary to enter the following password:

100	Saves the currently valid values of the parameters Use with the WPA and SEP commands

Meaning of the color highlight in the parameter table:

Gray:	The value of the parameter is loaded from the ID chip of the mechanics (p. 53).
Colorless:	The value of the parameter can only be changed via a command (SPA, SEP; special commands: AOS, CMO, RTR, WOS, WTR) or by using the corresponding operating elements of the PC software (p. 265).

Parameter ID (hexa-decimal)	Data type	Command level for write access	Element type	Parameter name	Description
0x02000200	Float	1	Input signal channel	Sensor Mech. Correction 1	Coefficients of the polynomial for mechanics linearization Refer to "Processing Input Signal Channels" (p. 21) and "ID Chip Detection" (p. 53) for details. Refer to "Referencing" (p. 46) and "Autozero Procedure for Compensating Weight Force" (p. 49) for further details on the ***Sensor Mech. Correction 1*** parameter.
0x02000300	Float	1	Input signal channel	Sensor Mech. Correction 2	
0x02000400	Float	1	Input signal channel	Sensor Mech. Correction 3	
0x02000500	Float	1	Input signal channel	Sensor Mech. Correction 4	
0x02000600	Float	1	Input signal channel	Sensor Mech. Correction 5	
0x02001000	INT	1	Input signal channel	Sensor Reference Signal Inversion	Inversion of the reference switch signal Refer to "Referencing" (p. 46) for details.
0x03000100	Float	1	Input signal channel	Sensor Elec. Correction 1	Coefficients of the polynomial for electronics linearization Refer to "Processing Input Signal Channels (p. 21) for details.
0x03000200	Float	1	Input signal channel	Sensor Elec. Correction 2	
0x03000300	Float	1	Input signal channel	Sensor Elec. Correction 3	
0x03000400	Float	1	Input signal channel	Sensor Elec. Correction 4	

Parameter ID (hexa-decimal)	Data type	Command level for write access	Element type	Parameter name	Description
0x03000500	Float	1	Input signal channel	Sensor Elec. Correction 5	
0x05000000	INT	1	Input signal channel	Digital Filter Type	Settings for digital filtering of the input signal channels Refer to "Processing Input Signal Channels" (p. 21) for details.
0x05000001	Float	1	Input signal channel	Digital Filter Bandwidth	
0x06000500	INT	1	Axis	ADC Channel For Target	Settings for the use of an analog input as a control source Refer to "Analog Input Signals" (p. 94) for details
0x06000501	Float	1	Axis	Analog Target Offset	
0x06010000	Float	1	Axis	Profile Generator Maximum Acceleration	Settings for the profile generator Refer to "Generating a Dynamics Profile" (p. 30) for details. Refer to "Generating Control Values" (p. 24) for further details on the ***Profile Generator Maximum Velocity*** parameter.
0x06010100	Float	1	Axis	Profile Generator Maximum Jerk	
0x06010300	INT	1	Axis	Profile Generator Enable	
0x06010400	Float	1	Axis	Profile Generator Maximum Velocity	
0x06010600	Float	1	Axis	Profile Generator Maximum Jounce	
0x07000000	Float	1	Axis	Position Range Limit min	Limits of the permissible range for target value or control value Refer to "Generating Control Values" (p. 24) for details.
0x07000001	Float	1	Axis	Position Range Limit max	
0x07000005	Float	1	Axis	Force Range Limit min	
0x07000006	Float	1	Axis	Force Range Limit max	
0x07000201	Float	1	Axis	Open Loop Slew-Rate	Maximum slew rate of the force in open-loop operation (in N/s)
0x07000300	Float	1	Axis	Position Servo P Term	Servo control parameter for the different control algorithms Refer to "Servo Algorithm and Other Control Value Corrections" (p. 33) for details.
0x07000301	Float	1	Axis	Position Servo I Term	
0x07000302	Float	1	Axis	Position Servo D Term	
0x07000307	Float	1	Axis	Velocity Servo P Term	
0x07000308	Float	1	Axis	Velocity Servo I Term	
0x07000309	Float	1	Axis	Velocity Servo D Term	

Parameter ID (hexa-decimal)	Data type	Command level for write access	Element type	Parameter name	Description
0x0700030A	Float	1	Axis	Force Servo P Term	
0x0700030B	Float	1	Axis	Force Servo I Term	
0x0700030C	Float	1	Axis	Force Servo D Term	
0x0700030D	Float	1	Axis	Force Servo P Term Floating	
0x0700030E	Float	1	Axis	Force Servo I Term Floating	
0x0700030F	Float	1	Axis	Force Servo D Term Floating	
0x07000310	Float	1	Axis	Gain Correction of Force Servo P Term Floating	
0x07000311	Float	1	Axis	FFC Position On Control Output	Gain values for the feedforward components refer to "Servo Algorithm and Other Control Value Corrections" (p. 33) for details
0x07000312	Float	1	Axis	FFC Velocity On Subordinate Velocity	
0x07000313	Float	1	Axis	FFC Acceleration On Control Output	
0x07000314	Float	1	Axis	FFC Force On Control Output	
0x07000315	Float	1	Axis	FFC Force On Subordinate Position Control	
0x07000316	Float	1	Axis	FFC Jerk On Subordinate Velocity Control	
0x07000317	Float	1	Axis	FFC Jounce On Control Output	
0x07000400	Float	1	Axis	Input Channel for Force Control Feedback	Axis allocation of force sensors Refer to "Allocating Axes to Channels" (p. 16) for details.
0x07000401	Float	1	Axis	Force Sensor Surface Detection Level	Force thresholds and delay time for contact detection Refer to "Contact Detection in Force Control" (p. 43) for details.
0x07000402	Float	1	Axis	Force Sensor Surface Lost Level	
0x07000403	Float	1	Axis	Force Sensor Surface Lost Timing	
0x07000404	Float	1	Axis	Force Sensor Autozero Value	Value of the force sensor after the autozero procedure Refer to "Autozero Procedure for Compensating Weight Force" (p. 49) for details.

Parameter ID (hexa-decimal)	Data type	Command level for write access	Element type	Parameter name	Description
0x07000405	Float	1	Axis	Force Sensor Surface Detection Ratio	Determination of the force threshold for contact detection Refer to "Contact Detection in Force Control" (p. 43) for details.
0x07000406	INT	1	Axis	Force Servo Surface Detection Method	
0x07000500	Float	1	Axis	Position from Sensor 1	Coefficients of the input matrix ***Position from Sensor 5*** and ***Position from Sensor 6*** only for C-413.2GA and .20A Refer to "Allocating Axes to Channels" (p. 16) for details.
0x07000501	Float	1	Axis	Position from Sensor 2	
0x07000502	Float	1	Axis	Position from Sensor 3	
0x07000503	Float	1	Axis	Position from Sensor 4	
0x07000504	Float	1	Axis	Position from Sensor 5	
0x07000505	Float	1	Axis	Position from Sensor 6	
0x07000601	CHAR	0	Axis	Position Axis Unit	Unit symbols for the different control variables Refer to "Control Modes and Control Variables" (p. 28) for details.
0x07000603	CHAR	0	Axis	Velocity Axis Unit	
0x07000604	CHAR	0	Axis	Force Axis Unit	
0x07000800	INT	1	Axis	Power Up Servo Enable	Automatic switching on of the servo mode after the C-413 is switched on or rebooted 0 = Servo mode is not switched on automatically 1 = Servo mode is switched on automatically
0x07000802	INT	1	Axis	Power Up AutoZero Enable	Automatic execution of the autozero procedure Refer to "Autozero Procedure for Compensating Weight Force" (p. 49) for details.
0x07000806	INT	1	Axis	Power Up Reference Move Enable	Automatic execution of the referencing move Refer to "Referencing" (p. 46) for details.
0x07000900	Float	1	Axis	Position On Target Tolerance	Settings for determining the on-target state in the different control modes Refer to "On-Target State" (p. 45) for details.
0x07000901	Float	1	Axis	Position On Target Settling Time	
0x07000902	Float	1	Axis	Velocity On Target Tolerance	
0x07000903	Float	1	Axis	Velocity On Target Settling Time	

Parameter ID (hexa-decimal)	Data type	Command level for write access	Element type	Parameter name	Description
0x07000904	Float	1	Axis	Force On Target Tolerance	
0x07000905	Float	1	Axis	Force On Target Settling Time	
0x07000A00	Float	1	Axis	AutoZero Low Value	Settings for the autozero procedure Refer to "Autozero Procedure for Compensating Weight Force " (p. 49) for details.
0x07000A01	Float	1	Axis	AutoZero High Value	
0x07000A03	Float	1	Axis	AutoZero Result	
0x07001005	Float	1	Axis	Position Report Scaling	Settings for using an analog output to monitor the position, force or velocity of the axis Refer to "Analog Output Signals" (p. 110) for details
0x07001006	Float	1	Axis	Position Report Offset	
0x07001007	Float	1	Axis	Force Report Scaling	
0x07001008	Float	1	Axis	Force Report Offset	
0x07001009	Float	1	Axis	Velocity Report Scaling	
0x0700100A	Float	1	Axis	Velocity Report Offset	
0x07030100	INT	1	Axis	Closed-Loop Control Mode	Settings for selecting the control mode Refer to "Control Modes and Control Variables" (p. 28) for details.
0x07030101	INT	1	Axis	Available Closed-Loop Control Modes	
0x07030105	INT	1	Axis	Force Control Working Mode	Working mode of the moving part of the mechanics in force control Refer to "Contact Detection in Force Control" (p. 43) for details.
0x07030300	Float	1	Axis	Velocity For Reference Move	Velocity for referencing moves Refer to "Referencing" (p. 46) for details.
0x08000100	Float	1	Axis	Notch Frequency	Settings for the notch filters Refer to "Servo Algorithm and Other Control Value Corrections" (p. 33) for details.
0x08000101	Float	1	Axis	Notch Frequency	
0x08000200	Float	1	Axis	Notch Rejection	
0x08000201	Float	1	Axis	Notch Rejection	
0x08000300	Float	1	Axis	Notch Bandwidth	
0x08000301	Float	1	Axis	Notch Bandwidth	
0x09000000	Float	1	Axis	Driving Factor 1	Coefficients of the output

Parameter ID (hexa-decimal)	Data type	Command level for write access	Element type	Parameter name	Description
0x09000001	Float	1	Axis	Driving Factor 2	matrix ***Driving Factor 3*** and ***Driving Factor 4*** only for C-413.2GA and .20A Refer to "Allocating Axes to Channels" (p. 16) for details.
0x09000002	Float	1	Axis	Driving Factor 3	
0x09000003	Float	1	Axis	Driving Factor 4	
0x0A000003	INT	1	Output signal channel	Output type	Settings for using analog outputs Refer to "Analog Output Signals" (p. 110) for details.
0x0A000004	INT	1	Output signal channel	Output Index	
0x0A000010	Float	1	Output signal channel	DAC Coefficient 0	
0x0A000020	Float	1	Output signal channel	DAC Coefficient 1	
0x0C000000	Float	1	Output signal channel	Soft Limit min	Lower limit for the output value (channels 1 and 2: current in A; channels 3 and 4: voltage in V) Note: ***Force Range Limit min*** (0x07000005) specifies the smallest permissible control value of the axis (in N).
0x0C000001	Float	1	Output signal channel	Soft Limit max	Upper limit for the output value (channels 1 and 2: current in A; channels 3 and 4: voltage in V) Note: ***Force Range Limit max*** (0x07000006) specifies the largest permissible control value of the axis (in N).
0x0C001000	Float	1	Output signal channel	I2T Peak Current [A]	Settings for I2t monitoring Refer to "I2t Monitoring for Protecting the Mechanics" (p. 51) for details.
0x0C001001	Float	1	Output signal channel	I2T Peak Current Time [s]	
0x0C001002	Float	1	Output signal channel	I2T Nominal Current [A]	

Parameter ID (hexa-decimal)	Data type	Command level for write access	Element type	Parameter name	Description
0x0C001003	INT	1	Output signal channel	I2T Active	
0x0D000000	CHAR	2	System	Device S/N	Serial number of the C-413 refer to "Type Plate" (p. 9) for details.
0x0D000700	CHAR	3	System	Hardware Name	Model designation of the C-413, refer also to "Type Plate" (p. 9).
0x0D000800	INT	3	System	Controller Address	Address of the C-413 Only for information (C-413 cannot be networked)
0x0E000200	Float	1	System	Servo Update Time	Servo cycle time of the C-413 Refer to "Optimizing the Servo Cycle Time" (p. 139) for details
0x0E000B00	INT	3	System	Number of Input Channels	Number of available axes and channels. Refer to "Commandable Elements" (p. 13) for details. Refer to "Deactivating Axes" (p. 52) for further details on the ***Number of System Axes*** parameter.
0x0E000B01	INT	3	System	Number of Output Channels	
0x0E000B02	INT	1	System	Number of System Axes	
0x0E000B03	INT	3	System	Number of Sensor Channels	
0x0E000B04	INT	3	System	Number of Driver Channels	
0x0F000100	CHAR	1	Sensor channel	Stage Type	Information on the connected mechanics Refer to "ID Chip Detection" (p. 53) and the description of the CST? command (p. 162) for details.
0x0F000200	CHAR	1	Sensor channel	Stage Serial Number	
0x10000500	INT	0	Axis	Fast IF Axis Input Usage	Use of input data that arrives at the C-413 via data segment 1 of the SPI interface: 0 = Off (input data is not used) 1 = Target (input data is used as target value (closed-loop operation) or control value (open-loop operation) for the axis)

Parameter ID (hexa-decimal)	Data type	Command level for write access	Element type	Parameter name	Description
0x10000501	INT	0	System	Fast IF Data Type	Selection of the data type for the data that is sent via data segment 1 of the SPI interface. If the "FLOAT 32 bit" data type is selected, the data can be written and read in the unit of the axis directly. Scaling is then not necessary. If an integer data type is selected, scaling is necessary. In this case, the parameters must also be set for the lower and upper limit of the value range (***Fast IF Data Low Limit*** 0x10000502, ***Fast IF Data High Limit*** 0x010000503). 0 = FLOAT 32 bit 1 = UINT 16 bit 2 = UINT 24 bit 3 = UINT 32 bit
0x10000502	Float	0	Axis	Fast IF Data Low Limit	Lower limit value for the data that is sent via data segment 1 of the SPI interface. Required for internal data scaling if an integer data type is selected (***Fast IF Data Type*** 0x10000501). The limit value is not evaluated for the "FLOAT 32 bit" data type.
0x10000503	Float	0	Axis	Fast IF Data High Limit	Upper limit value for the data that is sent via data segment 1 of the SPI interface. Required for internal data scaling if an integer data type is selected (***Fast IF Data Type*** 0x10000501). The limit value is not evaluated for the "FLOAT 32 bit" data type.

Parameter ID (hexa-decimal)	Data type	Command level for write access	Element type	Parameter name	Description
0x11000400	INT	0	System	UART Baudrate	Baud rate for the UART of the USB interface Refer to "Communication Interfaces" (p. 54) for details
0x13000004	INT	3	System	Maximum Number of Wave Points	Settings for the wave generator Refer to "Wave Generator" (p. 120) for details.
0x13000109	INT	1	Wave generator (= axis)	Wave Generator Table Rate	
0x1300010A	INT	3	System	Number of Wave Tables	
0x1300010B	Float	1	Wave generator (= axis)	Wave Offset	
0x16000000	INT	0	System	Data Recorder Table Rate	Settings for the data recorder Refer to "Configuring the Data Recorder" (p. 83) for details.
0x16000100	INT	3	System	Max Number of Data Recorder Channels	
0x16000200	INT	3	System	Data Recorder Max Points	
0x16000300	INT	0	System	Data Recorder Channel Number	
0xFFFF0001	INT	3	System	Firmware Valid/Invalid Mark	Information on the firmware and hardware of the C-413
0xFFFF0002	INT	3	System	CRC-32 of Firmware Program Code	
0xFFFF0003	INT	3	System	CRC-32 of Firmware Description	
0xFFFF0004	INT	3	System	Version of Firmware Description	
0xFFFF0006	CHAR	3	System	Unique Firmware Name	
0xFFFF0007	CHAR	3	System	Unique Board Name	
0xFFFF0008	INT	3	System	Version of Firmware	
0xFFFF000B	INT	3	System	Maximum Size of Flash	
0xFFFF000C	CHAR	3	System	Logical Device	
0xFFFF000D	CHAR	3	System	Description of Firmware	
0xFFFF000E	CHAR	3	System	Date of Firmware Development	
0xFFFF000F	CHAR	3	System	Name of Firmware Developer	

Parameter ID (hexa-decimal)	Data type	Command level for write access	Element type	Parameter name	Description
0xFFFF0010	INT	3	System	Length of Firmware	
0xFFFF0011	INT	3	System	Firmware Compatibility Index	
0xFFFF0012	INT	3	System	Relative Address from FW-Description to FW-Start	
0xFFFF0013	CHAR	3	System	Logical Device Type	
0xFFFF0014	INT	2	System	Hardware Revision of Board	
0xFFFF0015	INT	3	System	Execution Address of Firmware	
0xFFFF0016	INT	3	System	Configuration Options	

10 Maintenance

10.1 Cleaning the C-413

NOTICE

Short circuits or flashovers!

The C-413 contains electrostatic-sensitive devices that can be damaged by short-circuiting or flashovers when cleaning fluids penetrate the housing.

➢ Before cleaning, disconnect the C-413 from the power source by removing the mains plug.

➢ Prevent cleaning fluid from penetrating the housing.

➢ When necessary, clean the surfaces of the C-413's housing using a cloth dampened with a mild cleanser or disinfectant.

10.2 Updating Firmware

INFORMATION

Among other things, the *IDN? command reads the model designation of the C-413 and version number of the firmware.

Example of a C-413 response:

(c)2019-2020 Physik Instrumente (PI) GmbH & Co. KG,C-414.13030,01190196608,1.004

- C-414.13030: Model designation
- 1.004: Firmware version

Getting current firmware of the C-413

1. Query the C-413's model designation and the firmware version with the `*IDN?` command (e.g., in PITerminal or the ***Command entry*** window of PIMikroMove).
2. Contact our customer service department (p. 287) and specify the queried model designation and firmware version to obtain a current firmware version and associated information.

Requirements

✓ You have connected the C-413 to the PC via the USB interface or via TCP/IP (p. 65).

✓ The ***PIFirmwareManager*** program is installed on the PC (p. 59).

✓ You have copied the new firmware file, which you have received from our customer service department, to a directory on the PC.

- ✓ You have read and understood the documentation which you received from our customer service department together with the new firmware. You have learned from the documentation whether new parameters are introduced with the firmware update or the memory management of the C-413 changes.
- ✓ You have established communication with PIMikroMove or PITerminal between the C-413 and the PC (p. 72).

Updating the firmware of the C-413

- ➢ Start the ***PIFirmwareManager*** program on the PC and update the controller firmware.

  Proceed as described in the user manual SM164E (p. 2).

If new parameters have been introduced with the firmware update:

- ➢ If necessary, adapt the settings of the new parameters to your application (p. 265).

11 Troubleshooting

Fault: Mechanics do not move	
Possible causes	**Solution**
Cable not connected correctly	➢ Check the cable connections. ➢ If a connection assignment is given on the labels of the C-413 and/or mechanics, observe this assignment when connecting the mechanics (p. 64).
The mechanics were connected to the switched-on C-413	The sensor electronics in the mechanics have not been initialized, and the ID chip of the sensor (p. 53) has not been read out. ➢ Switch the C-413 off and on again, or reboot the C-413 with the `RBT` command or with the corresponding functions of the PC software.
Unsuitable cable used for the mechanics	If unsuitable cables are used, interferences can occur in the signal transmission between the mechanics and the C-413. ➢ Only use original PI parts to connect the mechanics to the C-413. The maximum cable length is 1 m. ➢ If you need longer cables, contact our customer service department (p. 287).
C-413 is defective	➢ Send the `ERR?` command and check the error code this returns. When error code 333 (internal hardware error) is reported back: 1. Switch the C-413 off and on again. 2. Get the error code again. 3. If error code 333 is still reported back, switch off the C-413 and contact our customer service department (p. 287).
Mechanics or cable is defective	➢ If available, replace the defective mechanics with different mechanics of the same type and test the new combination. Prevent damage to the mechanics as follows: ➢ Prevent the mechanics from overheating by activating I2t monitoring (p. 24). ➢ Prevent the mechanics from exceeding the maximally permissible operating frequency, e.g., by using suitable waveforms with the wave generator output (p. 120). ➢ Prevent the mechanics from oscillating by setting the control parameters appropriately. ➢ Prevent the axis from moving to the hard stop at a high velocity (possible in open-loop operation or when the control variable is the velocity or the force).
Motor driver of the C-413 is deactivated due to overheating	Overheating of the motor driver in the C-413 ➢ Reduce the frequency of the wave generator output
Incorrect configuration	➢ Check the parameter settings of the C-413 with the `SPA?` (volatile memory) and `SEP?` (nonvolatile memory) commands or in the

Fault: Mechanics do not move	
Possible causes	**Solution**
	Device Parameter Configuration window of PIMikroMove. Refer to "Adapting Settings" (p. 257) for details on parameter settings.
Incorrect command or incorrect syntax	➢ Send the `ERR?` command and check the error code this returns. ➢ Make sure that the used motion commands match the control mode (p. 28) and the servo mode (p. 22).
Motion commands or wave generator output are ignored.	➢ Send the `ERR?` command and check the error code this returns. ➢ Pay attention to the different priorities of the control sources; refer to "Generating Control Values" (p. 24).
Incorrect axis commanded	An axis identifier is even required in commands on systems with only one axis. ➢ Make sure that the correct axis identifier (p. 13) is used and that the commanded axis belongs to the correct mechanics.
In the case of analog control, there is no connection between the axis and the analog input.	➢ To control an axis via an analog input, allocate the corresponding input signal channel to the axis. Refer to "Analog Input Signals" (p. 94) for further information. ➢ When you have stopped the motion with `STP` or `#24`: Repeat the allocation.

Fault: Mechanics move unintentionally	
Possible causes	**Solution**
C-413 configuration	The C-413 can be configured with parameter settings so that the referencing move (p. 46) and/or the autozero procedure (p. 49) is run automatically after switching on or rebooting. ➢ Check the settings of the ***Power Up Reference Move Enable*** parameter (ID 0x07000806) and the ***Power Up AutoZero Enable*** parameter (ID 0x07000802) and adjust them if necessary.
C-413 was switched off	➢ Compensate for the lack of self-locking of the voice coil drive of the mechanics: – Avoid the overflow state (p. 191) of the axis (servo mode is switched off automatically when the axis has been in the overflow state for more than 60 s). – When the motion axis is aligned vertically: Run an autozero procedure (p. 49) for the axis so that the weight force of the moving mass is also compensated when servo mode is switched off. – Before switching off or rebooting the C-413, take suitable precautionary measures to ensure that no unexpected motion is possible due to lack of self-locking of the voice coil drive. Optimal sequence of the steps for starting and operating the C-
C-413 was rebooted (with `RBT` or corresponding functions of the PC software)	
Servo mode for the axis was switched off	

Fault: Mechanics move unintentionally	
Possible causes	**Solution**
	413: 1. Do a referencing move 2. Switch on servo mode 3. When the ***AutoZero Result*** parameter has not yet been set suitably: Run an autozero procedure

Fault: Mechanics are oscillating or positioning inaccurately	
Possible causes	**Solution**
The load was changed. The control mode was changed.	➢ If the mechanics are oscillating (unusual operating noise), switch servo mode or the C-413 off immediately. ➢ Switch servo mode back on only after you have modified the servo control parameter settings. ➢ Check the values of the servo control parameters each time the control mode (p. 28) is changed.
The profile generator worsens the dynamic behavior of the axis.	➢ In the following cases, check whether deactivating the profile generator (p. 30) improves the dynamic behavior of the axis: – The wave generator is running for the axis (p. 120). – An analog input is used as the control source for the axis (p. 107).

Fault: Force measurement by force sensor does not work	
Possible causes	**Solution**
Force sensor is not allocated to the axis	Force sensors are directly allocated to the logical axes of the C-413 via the ***Input Channel For Force Feedback*** parameter (ID 0x07000400) (p. 16). ➢ Ensure the correct allocation via the corresponding parameter setting.

Fault: Servo mode was switched off automatically	
Possible causes	**Solution**
Axis was in the overflow state for more than 60 s	➢ Check why the overflow state (p. 191) occurred. Possible causes: – The axis has not yet been referenced (query with `FRF?`). – Axis oscillates – When the control variable is the position or the velocity: The axis is blocked by an obstacle. – When the control variable is the velocity or the force: The axis has

Fault: Servo mode was switched off automatically	
Possible causes	**Solution**
	reached the hard stop. ➢ Remedy the cause of the overflow state.

Fault: Communication with the controller does not work	
Possible causes	**Solution**
The wrong communication cable is used or it is defective	➢ If necessary, check whether the cable works on a fault-free system.
Baud rate not configured correctly	A USB UART module is used in the C-413 for the USB interface. To successfully establish communication via USB, the baud rates of the PC and C-413 must therefore be identical. When the PC software offers PC baud rate selection when communication is established via USB: ➢ Adapt the baud rate of the PC to the current baud rate of the C-413.
Loss of the communication due to excessive utilization of the C-413 processor	When the communication is faulty or has been terminated: 1. Reboot the C-413. 2. Make sure that the C-413 is not doing any time-intensive tasks. 3. Extend the servo cycle time; refer to "Optimizing the Servo Cycle Time of the C-413" (p. 139).
Another program is accessing the interface.	➢ Close the other program.
Problems with special software	➢ Check whether the system works with other software, such as a terminal program or a development environment. ➢ Test the communication by sending the `*IDN?` or `HLP?` command. ➢ Make sure that you end the commands with an LF (line feed). Exception: Single-character commands are not followed by a terminating character; refer to "GCS Syntax for Syntax Version 2.0" (p. 143).

Fault: The customer software does not function with the PI drivers	
Possible causes	**Remedial measures**
Incorrect combination of driver routines/VIs	➢ Check whether the system functions with a terminal program (e.g., PITerminal). If so: ➢ Read the information in the corresponding software manual and compare your program code with the sample code on the data storage device with the PI Software Suite.

Fault: The Device Parameter Configuration window is not available in PIMikroMove.	
Possible causes	**Solution**
NI LabVIEW Run-Time Engine was not installed	➢ Install NI LabVIEW Run-Time Engine, refer to "Doing Initial Installation" (p. 59).

If the problem that occurred with your system is not in the list above or cannot be solved as described, contact our customer service department (p. 287).

12 Customer Service

For inquiries and orders, contact your PI representative or send us an email (mailto:service@pi.de).

- If you have questions concerning your system, provide the following information:
  - Product and serial numbers of all products in the system
  - Firmware version of the controller (if applicable)
  - Version of the driver or the software (if applicable)
  - PC operating system (if applicable)

If possible: Take photographs or make videos of your system that can be sent to our customer service if requested.

13 Technical Data

Subject to change. You can find the latest product specifications on the product web page at www.pi.ws (https://www.physikinstrumente.com/en/).

13.1 Specifications

13.1.1 Data Table

	C-413.1G	C-413.20/.20A, C-413.2G/.2GA
Function	PIMag® motion controller for voice coil drives, 1 channel, housed device	PIMag® motion controller for voice coil drives, 2 channels C-413.20/.20A: OEM board C-413.2G/.2GA: Housed device
Motor channels	1	2
Sensor channels	2	4

Motion and servo controller	C-413.1G	C-413.20/.20A, C-413.2G/.2GA
Controller type	PID controller for force, position and velocity; parameter changing during operation	PID controller for force, position and velocity; parameter changing during operation
Servo cycle time	100 µs to 200 µs, selectable in 4 steps	100 µs to 200 µs, selectable in 4 steps
Profile generator	Trapezoidal velocity profile, specification of the maximum velocity and acceleration	Trapezoidal velocity profile, specification of the maximum velocity and acceleration
Encoder input	SPI sensor interface	SPI sensor interface
Reference switch	2 × TTL, direction-sensing	4 × TTL, direction-sensing

Electrical properties	C-413.1G	C-413.20/.20A, C-413.2G/.2GA
Max. output voltage	24 V	24 V
Max. output current	±1.5 A (regulated)	±1.5 A (regulated)

Interfaces and operation	C-413.1G	C-413.20/.20A, C-413.2G/.2GA
Communication interfaces	TCP/IP	USB 2.0, real time SPI
Motor / sensor connector	D-sub 9 (f) for motor, D-sub 25 (f) for sensor	D-sub 15 (f) combined for motor and sensor
I/O port	2 × analog output, -10 to 10 V, 17 bit, 1 kHz	2 × analog input, -10 to 10 V, 16 bit, 1 kHz (only .20A and .2GA)

Interfaces and operation	C-413.1G	C-413.20/.20A, C-413.2G/.2GA
	4 × digital input, 24 V 6 × digital output, 24 V	2 × analog output, -10 to 10 V, 17 bit, 1 kHz (only .20A and .2GA) 6 × digital outputs (open collector, voltage range 5 V to 24 V, 33 kΩ internal pull-up to 5 V) 4 × digital input (5 V TTL level, to 24 V max. input voltage, 10 kΩ input resistance)
Command set	PI General Command Set (GCS)	PI General Command Set (GCS)
User software	PIMikroMove	PIMikroMove
Application programming interfaces	C, C++, C#, MATLAB, NI LabVIEW, Python	C, C++, C#, MATLAB, NI LabVIEW, Python
Supported functions	Point-to-point motion. Data recorder. Wave generator. Autozero.ID chip detection.	Point-to-point motion. Data recorder. Wave generator. Autozero. ID chip detection. I^2t monitoring.

Miscellaneous	C-413.1G	C-413.20/.20A, C-413.2G/.2GA
Operating voltage	24 V DC from external power adapter (included in the scope of delivery)	24 V DC from external power adapter (included in the scope of delivery for C-413.2G and .2GA)
Max. current consumption	2 A	2 A
Operating temperature range	5 to 50°C	5 to 50 °C
Mass	0.3 kg	0.3 kg
Dimensions	210 mm × 28 mm × 105 mm	189 mm × 28 mm × 105 mm (.2G/.2GA) 160 mm × 18 mm × 100 mm (.20/.20A)

13.1.2 Maximum Ratings

The C-413 is designed for the following maximum ratings:

Input on:	Maximum operating voltage ⚠	Operating frequency ⚠	Maximum current consumption ⚠
M8 panel plug, 4-pin (m)	24 V	⎓	2 A

13.1.3 Ambient Conditions and Classifications

The following ambient conditions and classifications for the C-413 must be observed:

Area of application	For indoor use only
Maximum altitude	2000 m
Air pressure	1100 hPa to 0.1 hPa
Relative humidity	Highest relative humidity 80 % for temperatures up to 31 °C Decreasing linearly to 50 % relative air humidity at 40 °C
Storage temperature	0 °C to 70 °C
Transport temperature	–25 °C to +85 °C
Overvoltage category	II
Protection class	I
Degree of pollution	2
Degree of protection according to IEC 60529	IP20

13.2 Dimensions

Dimensions in mm. Note that the decimal points are separated by a comma in the drawings.

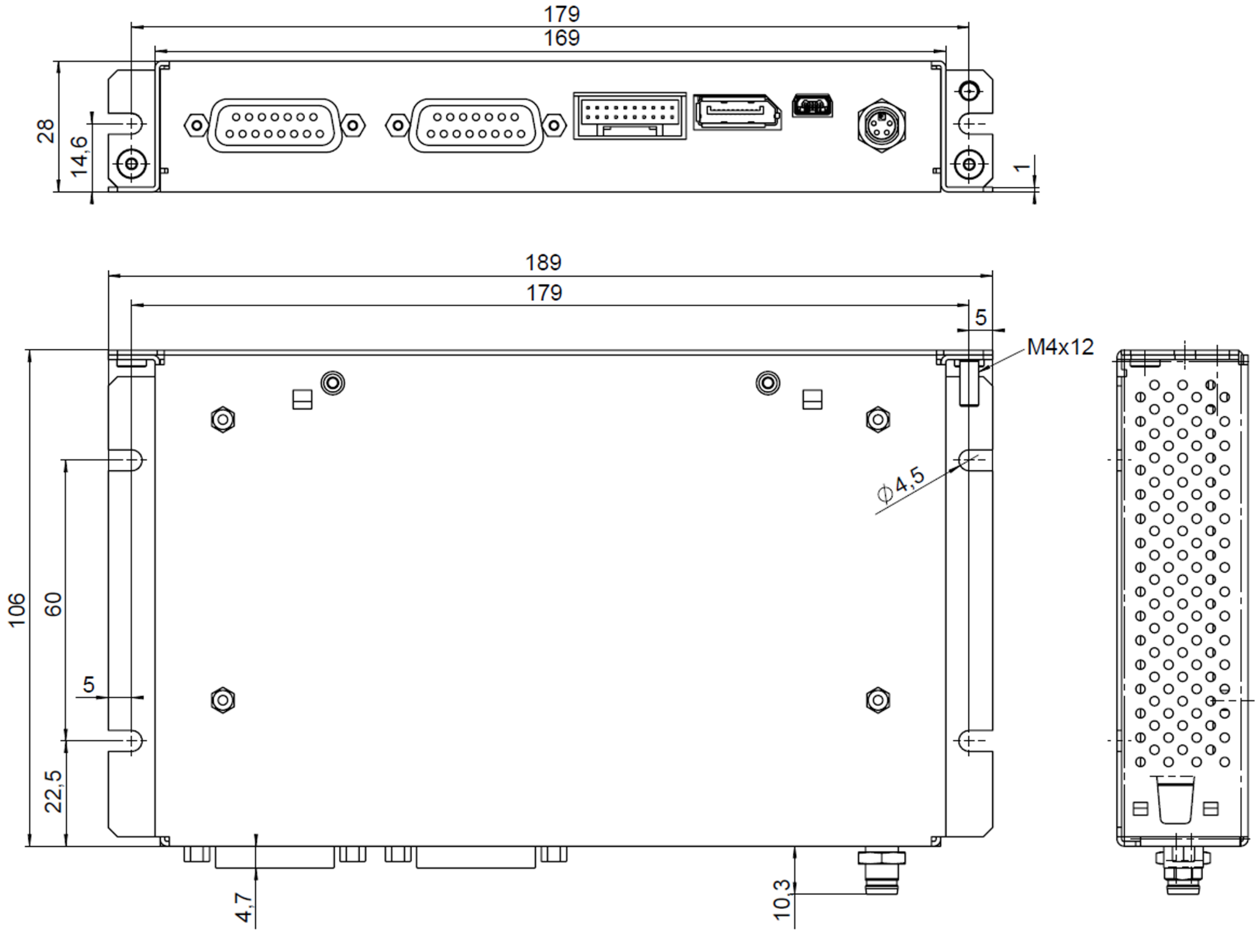


Figure 40: Dimensions of the C-413.2GA and C-413.2G models

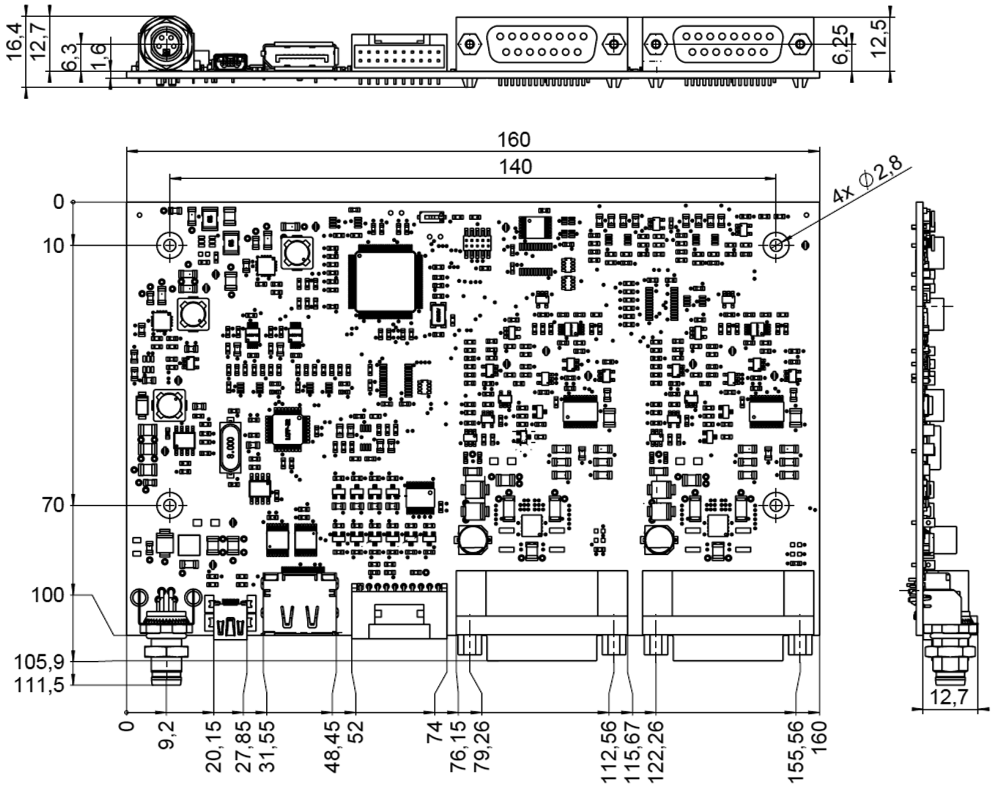
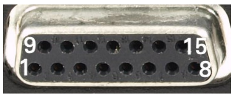


Figure 41: Dimensions of the C-413.20A and C-413.20 models

13.3 Pin Assignment

13.3.1 Motor & Sensor

D-sub socket, 15-pole, female



Pin	Signal	Direction	Function***
1	REF2*/4**	Input	Reference switch (direction sensing) for input signal channel 2* / 4**, TTL
2	Motor N1*/2**	Output	Output current for output signal channel 1* / 2** (±1.5 A, regulated; 24 V max.)
3	GND	-	GND
4	5 V	Output	Output voltage, maximum output power 500 mW
5	CS_MEM2*/4**	Output	SPI Chip Select: Selection of the ID chip for input signal channel 2* / 4** for data transmission
6	CS_MEM1*/3**	Output	SPI Chip Select: Selection of the ID chip for input signal channel 1* / 3** for data transmission
7	SEN_MOSI1*/2**	Output	SPI data line for ID chip and sensor signals of the input signal channels 1 and 2* / 3 and 4**
8	SEN_MISO1*/2**	Input	SPI data line for ID chip and sensor signals of the input signal channels 1 and 2* / 3 and 4**
9	Motor P1*/2**	Output	Output current for output signal channel 1* / 2** (±1.5 A, regulated; 24 V max.)
10	GND	-	GND
11	GND	-	GND
12	CS_SEN2*/4**	Output	SPI Chip Select: Selection of input signal channel 2* / 4** for data transmission
13	REF1*/3**	Input	Reference switch (direction sensing) for input signal channel 1* / 3**, TTL
14	SEN_CLK1*/2**	Output	SPI Serial Clock for ID chip and sensor signals of input signal channels 1 and 2* / 3 and 4**
15	CS_SEN1*/3**	Output	SPI Chip Select: Selection of input signal channel 1* / 3** for data transmission

* Assignment for **Motor & Sensor 1** socket

** Assignment for **Motor & Sensor 2** socket

******* "Input signal channel" corresponds to "sensor" here; "output signal channel" corresponds to "drive" here

13.3.2 I/O

PUD panel plug, 20-pin, male

Figure 42: Front view of the PUD panel plug

Pin	Signal	Function	Pin	Signal	Function
1	AIN1	Analog input, input signal channel 5* -10 to 10 V, 16 bit, 1 kHz	2	GND	GND
3	AIN2	Analog input, input signal channel 6* -10 to 10 V, 16 bit, 1 kHz	4	GND	GND
5	AOUT1	Analog output, output signal channel 3* -10 to 10 V, 17 bit, 1 kHz	6	GND	GND
7	AOUT2	Analog output, output signal channel 4* -10 to 10 V, 17 bit, 1 kHz	8	GND	GND
9	DIN1	Digital input 1***	10	DIN2	Digital input 2***
11	DIN3	Digital input 3***	12	DIN4	Digital input 4***
13	5 V	Output voltage, maximum output power 500 mW	14	GND	GND
15	DOUT1	Digital output 1**	16	DOUT2	Digital output 2**
17	DOUT3	Digital output 3**	18	DOUT4	Digital output 4**
19	DOUT5	Digital output 5**	20	DOUT6	Digital output 6** Not accessible for commands; output of the servo cycles

* Analog inputs and outputs only with C-413.2GA and .20A

** Digital outputs: Open collector (range 5 V to 24 V, 33 kOhm internal pull-up to 5 V)

*** Digital inputs: 5 V TTL level, up to 24 V max. input voltage, 10 kOhm input resistance

13.3.3 C-413.1IO Cable for the I/O Connection

JST connector, 20-pin, female, open end

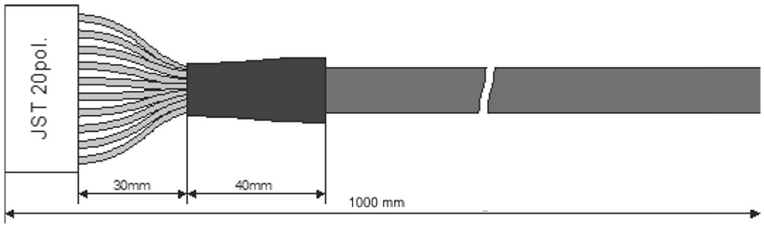


Figure 43: C-413.1IO cable

Specifications

Temperature range: -25 °C to +85 °C
Nominal current: 1 A AC/DC
Insulation resistance: 50 MΩ min.
Nominal voltage: 50 V AC/DC
Voltage impulse: 500 V AC for 1 minute

Pin	Wire Color	Function at the I/O Panel Plug of the C-413
1	Black	Analog input, input signal channel 5* -10 to 10 V, 16 bit, 1 kHz
2	Brown	GND
3	Red	Analog input, input signal channel 6* -10 to 10 V, 16 bit, 1 kHz
4	Orange	GND
5	Yellow	Analog output, output signal channel 3* -10 to 10 V, 17 bit, 1 kHz
6	n.c.	GND
7	Green	Analog output, output signal channel 4* -10 to 10 V, 17 bit, 1 kHz
8	n.c.	GND
9	Blue	Digital input 1***
10	Violet	Digital input 2***
11	Gray	Digital input 3***
12	White	Digital input 4***
13	White/black	5 V output voltage, maximum output power 500 mW
14	White/brown	GND
15	White/red	Digital output 1**
16	White/orange	Digital output 2**
17	White/yellow	Digital output 3**

Pin	Wire Color	Function at the I/O Panel Plug of the C-413
18	White/green	Digital output 4**
19	White/blue	Digital output 5**
20	White/violet	Digital output 6** Not accessible for commands; output of the servo cycles

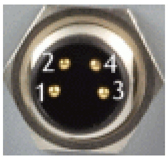
* Analog inputs and outputs only with C-413.2GA and .20A

** Digital outputs: Open collector (range 5 V to 24 V, 33 kOhm internal pull-up to 5 V)

*** Digital inputs: 5 V TTL level, up to 24 V max. input voltage, 10 kOhm input resistance

13.3.4 Power Supply Connector 24 V DC

Phoenix M8 panel plug, 4-pole, male



Pin	Function
1	GND (power)
2	GND (power)
3	Input: 24 V DC
4	Input: 24 V DC

14 Old Equipment Disposal

In accordance with EU law, electrical and electronic equipment may not be disposed of in EU member states via the municipal residual waste.

Dispose of your old device according to international, national, and local rules and regulations.

To fulfill the responsibility as the product manufacturer, Physik Instrumente (PI) SE & Co. KG undertakes environmentally correct disposal of all old PI equipment made available on the market after 13 August 2005 without charge.

If you have an old device from PI, you can send it to the following address free of charge:

Physik Instrumente (PI) SE & Co. KG

Auf der Römerstraße 1

76228 Karlsruhe, Germany